

An efficient goal-oriented sampling strategy using reduced basis method for parametrized elastodynamic problems

K. C. Hoang¹, P. Kerfriden¹, B. C. Khoo², and S. P. A. Bordas^{1,3}

¹School of Engineering, Cardiff University, The Parade, CF24 3AA Cardiff, United Kingdom

²Singapore MIT-Alliance, National University of Singapore, Singapore

³Faculté des Sciences, de la Technologie et de la Communication, University of Luxembourg, 6 rue Richard Coudenhove-Kalergi, L-1359 Luxembourg

Emails: {hoangkhacchi, pierre.kerfriden, stephane.bordas}@gmail.com, mpekbc@nus.edu.sg

Abstract

In this paper, we study the class of linear elastodynamic problems with affine parameter dependence using a goal-oriented approach by finite element (FE) and reduced basis (RB) methods. The main contribution of this paper is the “goal-oriented” proper orthogonal decomposition (POD)–Greedy sampling strategy within the RB approximation context. The proposed sampling strategy looks for the parameter points such that the output error approximation will be minimized by Greedy iterations. In estimating such output error approximation, the standard POD–Greedy algorithm is invoked to provide enriched RB approximations for the FE outputs. We propose a so-called “cross-validation” process to choose adaptively the dimension of the enriched RB space corresponding with the dimension of the RB space under consideration. Numerical results show that the new goal-oriented POD–Greedy sampling procedure with the cross-validation process improves significantly the space-time output computations in comparison with the ones computed by the standard POD–Greedy algorithm. The method is thus ideally suited for repeated, rapid and reliable evaluations of input-output relationships in the space-time setting.

Keywords: reduced basis method; goal-oriented POD–Greedy algorithm; wave equation; goal-oriented asymptotic error; cross-validation

1 Introduction

The design, optimization and control procedures of engineering problems often require several forms of performance measures or outputs – such as displacements, heat fluxes or flowrates. Generally, these outputs are functions of field variables such as displacements, temperature or velocities which are usually governed by a partial differential equation (PDE). The parameter or input will frequently define a particular configuration of the model problem. Therefore, the relevant system behavior will be described by an implicit input-output relationship; where its computation requires the solution of the underlying parameter-PDE (or μ PDE). We pursue model order reduction (MOR) methods (i.e., snapshots-POD [1, 2, 3, 4, 5, 6] and RB [7, 8]) which permits the *efficient* and *reliable* evaluation of this PDE-induced input-output relationship in *many query* and *real-time* contexts.

The RB method was first introduced in the late 1970s for nonlinear analysis of structures and has been investigated and developed more broadly [9]. Recently, the RB method was well developed for various kinds and classes of parametrized PDEs such as: the eigenvalue problems [10], the

coercive/non-coercive affine/non-affine linear/nonlinear elliptic PDEs [7, 11], the coercive/non-coercive affine/non-affine linear/nonlinear parabolic PDEs [12, 13], the coercive affine linear hyperbolic PDEs [14, 8], and several highly nonlinear problems such as Burger’s equation [15, 16] and Boussinesq equation [17]. For the linear wave equation, the RB method and associated *a posteriori* error estimation was developed successfully with some levels [14, 18, 19]; however, in the RB context none of these works have focused on constructing optimally goal-oriented RB basis functions.

Goal-oriented error estimates in the context of FE analysis have been investigated deeply and widely [20, 21, 22, 23, 24, 25, 26, 27] (we only cite some typical works as there many on this topic). For the wave equation, the most well-known method is the dual-weighted residual (DWR) one which was proposed by Rannacher and co-workers [28, 29, 30, 31]. In those works, the authors quantified the *a posteriori* error of the interest output in order to finer locally the finite element mesh in an adaptive manner. The final goal is to minimize computational efforts and maximize the accuracy of the interest output in an adaptive and controllable manner. In particular, the DWR method makes use of an auxiliary dual (or sensitivity) equation to derive an *a posteriori* error expression for the interest output from the primal residual and the dual solution of that dual equation in space-time setting.

Goal-oriented error estimates in the context of MOR is currently an active research topic and has been investigated by several authors. In this regard, the construction procedure of these goal-oriented MOR basis functions is the key issue. For instance, Liu et al. [32] used a Greedy algorithm to construct goal-oriented RB basis functions based on asymptotic output errors [33]; the surrogate RB model was then used in an inverse analysis. Chen et al. [34] developed hybrid and goal-oriented Greedy sampling algorithms to compute failure probability for PDEs with random input data. In another work, Urban et al. [35] developed a goal-oriented sampling strategy which consists of solving an optimization problem and a goal-oriented Greedy sampling to find the optimal parameter samples to best approximate interest outputs. We note that the representative works mentioned above are for stationary and steady problems.

For dynamic problems, goal-oriented sampling strategy for MOR were also addressed by several authors. Meyer and Matthies [36] combined the DWR with the snapshots-POD method to solve a nonlinear dynamics problem. They quantified the *a posteriori* error approximation from the contributions of all POD snapshots; then the MOR basis functions are built (based on these POD snapshots) by keeping only the snapshots that caused large errors and removing all the ones which caused smaller errors. In another well-known approach by Bui [37] and Willcox [38], they solved a PDE-constrained optimization problem to find the optimally goal-oriented set of basis functions. In this way, the optimally goal-oriented basis functions are found such that they minimize the true output errors (with appropriate regularization techniques) and subject to equilibrium PDE-constraints [39, 40].

In general, those two above approaches are optimal. However, their computational cost are very expensive since one has to compute all the FEM solutions/outputs over the entire parameter domain (all POD snapshots – for the former approach), and in every iteration within optimization solvers (for the latter approach); and hence, it would limit the number of input parameters in comparison with the RB approach.

In this work, we aim to build an optimally goal-oriented set of MOR basis functions without computing and storing all the POD snapshots as the aforementioned approaches. The best way to do that is by using the RB method with Greedy sampling strategy (see, for instance [11, 7, 8]). Thanks to the Greedy iterations, the proposed algorithm now looks for the parameter points such that the output error approximation will be minimized. For the linear wave equation, this idea is novel and further develops the idea of the standard POD–Greedy sampling procedure currently used [41, 17, 8], where the algorithm will pick up optimally all parameter points such that the error (or error indicator) of the field variable is minimized. By this way, we expect to improve significantly the accuracy of the RB output functional computations; but consequently, we might lose the rapid convergence rate of the field variable as in the standard POD–Greedy algorithm. In fact, as we can see later in the numerical results section, the convergent rate of the field variable by the two algorithms are quite similar; while

the convergent rate of the output by the goal-oriented POD–Greedy algorithm is faster than that of the standard POD–Greedy one^{*}.

In particular, the output error approximation used in this work is a kind of the asymptotic output error (such as in [32, 33]) where the FE outputs will be approximated by the enriched RB outputs. The standard POD–Greedy algorithm will be invoked to compute such enriched RB outputs. Heuristically, the dimension of the enriched RB space will be usually set to two times larger than that of the RB space under consideration (see [32, 33]). In this work, however, we devise a simple yet efficient algorithm called “cross-validation” process to find out adaptively the enriched RB dimension corresponding with each RB dimension under consideration. That process is performed within the offline stage of the proposed goal-oriented algorithm. We also note that this output error approximation will be used as both offline and online error indicators in the offline and online computational stages, respectively.

The potential context for the proposed goal-oriented algorithm is described as follows. Suppose that one considers the linear parametrized wave equation with several different quantities of interest, and one wants to estimate the RB approximations of these quantities of interest. Note that these quantities of interest (or some of them) might be *a priori* unknown, i.e., they may exist at the time of consideration, or they can appear afterwards depending on one’s needs. Clearly, the standard POD–Greedy algorithm is not sufficient for this situation as it only provides the best approximations for the solution (or field variable) – and not for these quantities of interest. Our proposal is as follows. The standard algorithm is implemented *first* and only *once* to create standard RB spaces[†] to be used in the output error estimation of the proposed goal-oriented algorithm afterwards. Then, for each particular quantity of interest, the proposed goal-oriented algorithm will be performed once to build goal-oriented RB spaces corresponding to that quantity. By this way, the goal-oriented RB spaces are optimal for the quantity of interest under consideration (or in other words, best approximate this quantity of interest); and thus are much better than the standard RB spaces created by the standard algorithm.

The paper is organized as follows. In Section 2, we introduce necessary definitions, concepts and notations and then state the problem using a semidiscrete approach: fully discretizing in space using Galerkin FEM and marching in time using Newmark’s trapezoidal rule. In Section 3, we describe various topics related to the RB methodology: approximation, the standard versus goal-oriented algorithms, error estimations and offline-online computational procedure. In Section 4, we verify the performance of the proposed algorithm by investigating numerically two problems: a 2D linear elastodynamic problem and a 3D dental implant simulation problem [8]. Finally, we provide some concluding remarks in Section 5.

2 Problem statement

2.1 Abstract formulation

We consider a spatial domain $\Omega \in \mathbb{R}^d$ ($d = 1, 2, 3$) with Lipschitz continuous boundary $\partial\Omega$. We denote the Dirichlet portion of the boundary by $\Gamma_{D,i}$, $1 \leq i \leq d$. We then introduce the Hilbert spaces

$$Y^e = \{v \equiv (v_1, \dots, v_d) \in (H^1(\Omega))^d \mid v_i = 0 \text{ on } \Gamma_{D,i}, i = 1, \dots, d\}, \quad (1a)$$

$$X^e = (L^2(\Omega))^d. \quad (1b)$$

Here, $H^1(\Omega) = \{v \in L^2(\Omega) \mid \nabla v \in (L^2(\Omega))^d\}$ where $L^2(\Omega)$ is the space of square-integrable functions over Ω . We equip our spaces with inner products and associated norms $(\cdot, \cdot)_{Y^e}$ ($(\cdot, \cdot)_{X^e}$) and $\|\cdot\|_{Y^e} = \sqrt{(\cdot, \cdot)_{Y^e}}$ ($\|\cdot\|_{X^e} = \sqrt{(\cdot, \cdot)_{X^e}}$), respectively; a typical choice is

^{*}In subsequent sections, for simplicity we shall call the “standard algorithm” to mention the standard POD–Greedy algorithm [41], and the “goal-oriented algorithm” to mention the proposed goal-oriented POD–Greedy algorithm, respectively.

[†]We will see later that these standard RB spaces have quite high dimensions (higher than that of the goal-oriented RB spaces) but are still very small compared to the FE space dimension.

$$(w, v)_{Y^e} = \sum_{i,j} \int_{\Omega} \frac{\partial w_i}{\partial x_j} \frac{\partial v_i}{\partial x_j} + w_i v_i, \quad (2a)$$

$$(w, v)_{X^e} = \sum_i \int_{\Omega} w_i v_i, \quad (2b)$$

where the summation over spatial dimensions $1 \leq i, j \leq d$ is assumed throughout this paper.

We define an input parameter set $\mathcal{D} \in \mathbb{R}^P$, a typical point in which shall be denoted $\mu \equiv (\mu_1, \dots, \mu_P)$. We then define the parametrized bilinear forms a in Y^e , $a : Y^e \times Y^e \times \mathcal{D} \rightarrow \mathbb{R}$; m, c, f, ℓ are parametrized continuous bilinear and linear forms in X^e , $m : X^e \times X^e \times \mathcal{D} \rightarrow \mathbb{R}$, $c : X^e \times X^e \times \mathcal{D} \rightarrow \mathbb{R}$, $f : X^e \times \mathcal{D} \rightarrow \mathbb{R}$ and $\ell : X^e \rightarrow \mathbb{R}$.

The “exact” continuous problem is stated as follows: given a parameter $\mu \in \mathcal{D} \subset \mathbb{R}^P$, the field variable $u^e(x, t; \mu) \in Y^e$ satisfies the weak form of the μ -parametrized hyperbolic PDE (assume Rayleigh damping)

$$m \left(\frac{\partial^2 u^e(x, t; \mu)}{\partial t^2}, v; \mu \right) + c \left(\frac{\partial u^e(x, t; \mu)}{\partial t}, v; \mu \right) + a(u^e(x, t; \mu), v; \mu) = g(t)f(v; \mu), \quad \forall v \in Y^e, t \in [0, T], \mu \in \mathcal{D}, \quad (3)$$

with initial conditions: $u^e(x, 0; \mu) = 0$, $\frac{\partial u^e(x, 0; \mu)}{\partial t} = 0$.

In the above equation, x denotes the coordinate of a point in the domain Ω , t is the time variable, $[0, T]$ is a finite time interval and the explicit forms of a, m, c and f could be defined as: $\forall w, v \in Y^e, \mu \in \mathcal{D}$,

$$m(w, v; \mu) = \sum_i \int_{\Omega} \rho v_i w_i, \quad (4a)$$

$$c(w, v; \mu) = \sum_i \int_{\Omega} \alpha \rho v_i w_i + \sum_{i,j,k,l} \int_{\Omega} \beta \frac{\partial v_i}{\partial x_j} C_{ijkl} \frac{\partial w_k}{\partial x_l}, \quad (4b)$$

$$a(w, v; \mu) = \sum_{i,j,k,l} \int_{\Omega} \frac{\partial v_i}{\partial x_j} C_{ijkl} \frac{\partial w_k}{\partial x_l}, \quad (4c)$$

$$f(v; \mu) = \sum_i \int_{\Omega} b_i v_i + \sum_i \int_{\Gamma_N} v_i \phi_i. \quad (4d)$$

Where, ρ is the mass density; α is the mass-proportional Rayleigh damping coefficient; β is the stiffness-proportional Rayleigh damping coefficient; C_{ijkl} is the material elasticity tensor; b is a body force and ϕ is a surface traction applied to a region of the domain Ω ; $g(t)$ is the time history associated with the external loading $f(v; \mu)$; Γ_D and Γ_N are the Dirichlet and Neumann boundaries, respectively. We note that the input parameter μ could appear in (not limited to) either $\rho(\mu)$, $\alpha(\mu)$, $\beta(\mu)$, $C_{ijkl}(\mu)$ and even $b(\mu)$, $\phi(\mu)$ and $g(t; \mu)$.

We then evaluate a quantity of interest (output) from

$$s^e(\mu) = \int_0^T \int_{\Gamma_o} u^e(x, t; \mu) \Sigma(x, t) dx dt, = \int_0^T \ell(u^e(x, t; \mu)) dt. \quad (5)$$

Here, Γ_o are some (output) spatial regions of interest and $\Sigma(x, t)$ is an extractor which depends on the view position of an “observer” in the space-time domain; and $\ell(u^e(x, t; \mu)) = \int_{\Gamma_o} u^e(x, t; \mu) \Sigma(x, t) dx$.

We shall assume that the bilinear forms $a(\cdot, \cdot; \mu)$ and $m(\cdot, \cdot; \mu)$ are continuous,

$$a(w, v; \mu) \leq \gamma \|w\|_{Y^e} \|v\|_{Y^e} \leq \gamma_0 \|w\|_{Y^e} \|v\|_{Y^e}, \quad \forall w, v \in Y^e, \forall \mu \in \mathcal{D}, \quad (6a)$$

$$m(w, v; \mu) \leq \varrho \|w\|_{X^e} \|v\|_{X^e} \leq \varrho_0 \|w\|_{X^e} \|v\|_{X^e}, \quad \forall w, v \in Y^e, \forall \mu \in \mathcal{D}, \quad (6b)$$

coercive,

$$0 \leq \alpha_0 \leq \alpha(\mu) \equiv \inf_{v \in Y^e} \frac{a(v, v; \mu)}{\|v\|_{Y^e}^2}, \quad \forall \mu \in \mathcal{D}, \quad (7a)$$

$$0 \leq \sigma_0 \leq \sigma(\mu) \equiv \inf_{v \in Y^e} \frac{m(v, v; \mu)}{\|v\|_{X^e}^2}, \quad \forall \mu \in \mathcal{D}; \quad (7b)$$

and symmetric $a(v, w; \mu) = a(w, v; \mu), \forall w, v \in Y^e, \forall \mu \in \mathcal{D}$ and $m(v, w; \mu) = m(w, v; \mu), \forall w, v \in Y^e, \forall \mu \in \mathcal{D}$. (We (plausibly) suppose that $\gamma_0, \varrho_0, \alpha_0$ and σ_0 may be chosen independent of \mathcal{N} .) In addition, the linear forms $f(v) : Y^e \rightarrow \mathbb{R}$ and $\ell(v) : Y^e \rightarrow \mathbb{R}$ are assumed to be bounded with respect to $\|\cdot\|_{Y^e}$ and $\|\cdot\|_{X^e}$, respectively. Under these conditions, there exists a unique so-called “weak” (or “variational”) solution $u^e(x, t; \mu) \in Y^e$ of the equation (3) [31, 42].

We shall make an important assumption, that is, a, m, c and f depend affinely on the parameter μ and thus can be expressed as

$$m(w, v; \mu) = \sum_{q=1}^{Q_m} \Theta_m^q(\mu) m^q(w, v), \quad \forall w, v \in Y^e, \mu \in \mathcal{D}, \quad (8a)$$

$$c(w, v; \mu) = \sum_{q=1}^{Q_c} \Theta_c^q(\mu) c^q(w, v), \quad \forall w, v \in Y^e, \mu \in \mathcal{D}, \quad (8b)$$

$$a(w, v; \mu) = \sum_{q=1}^{Q_a} \Theta_a^q(\mu) a^q(w, v), \quad \forall w, v \in Y^e, \mu \in \mathcal{D}, \quad (8c)$$

$$f(v; \mu) = \sum_{q=1}^{Q_f} \Theta_f^q(\mu) f^q(v), \quad \forall v \in Y^e, \mu \in \mathcal{D}, \quad (8d)$$

for some (preferably) small integers $Q_{m,c,a,f}$. Here, the smooth functions $\Theta_{m,c,a,f}^q(\mu) : \mathcal{D} \rightarrow \mathbb{R}$ depend on μ , but the bilinear and linear forms m^q, c^q, a^q and f^q do *not* depend on μ .

2.2 Finite element discretization

We shall use the “method of lines” approach: fully discretize in space using Galerkin FE and discretize in time using Newmark’s trapezoidal scheme ($\gamma^N = \frac{1}{2}, \beta^N = \frac{1}{4}$). We introduce a reference finite element approximation space $Y \subset Y^e (\subset X^e)$ of dimension \mathcal{N} ; we further define $X \equiv X^e$. Note that Y and X shall inherit the inner product and norm from Y^e and X^e , respectively. For time integration: we divide $I = [0, T]$ into K subintervals of equal length $\Delta t = \frac{T}{K}$ and define $t^k = k\Delta t, 0 \leq k \leq K$. Recall that the Newmark’s trapezoidal scheme is implicit and unconditionally stable. Furthermore, Δt and the FE mesh size will be chosen such that they satisfy the solvability, stability and accuracy conditions following [43] (Chapter 9.4.4) or [44] (Chapter 9.1). Thus, our “true” finite element approximation $u(x, t^k; \mu) (\equiv u^k(\mu)) \in Y$ to the “exact” problem is equivalent to solving $(K - 1)$ following elliptic problems [44]:

$$\mathcal{A}(u^{k+1}(\mu), v; \mu) = \mathcal{F}(v), \quad \forall v \in Y, \mu \in \mathcal{D}, 1 \leq k \leq K - 1, \quad (9)$$

where [45]

$$\left\{ \begin{array}{l} \mathcal{A}(u^{k+1}(\mu), v; \mu) = \frac{1}{\Delta t^2} m(u^{k+1}(\mu), v; \mu) + \frac{1}{2\Delta t} c(u^{k+1}(\mu), v; \mu) + \frac{1}{4} a(u^{k+1}(\mu), v; \mu), \\ \mathcal{F}(v) = -\frac{1}{\Delta t^2} m(u^{k-1}(\mu), v; \mu) + \frac{1}{2\Delta t} c(u^{k-1}(\mu), v; \mu) - \frac{1}{4} a(u^{k-1}(\mu), v; \mu) \\ \quad + \frac{2}{\Delta t^2} m(u^k(\mu), v; \mu) - \frac{1}{2} a(u^k(\mu), v; \mu) + g^{eq}(t^k) f(v; \mu), \\ g^{eq}(t^k) = \frac{1}{4} g(t^{k-1}) + \frac{1}{2} g(t^k) + \frac{1}{4} g(t^{k+1}), \end{array} \right. \quad (10)$$

with initial conditions[‡]: $u^0(\mu) = 0$, $\frac{\partial u^0(\mu)}{\partial t} = 0$; we then evaluate the output of interest from (using the trapezoidal rule for integral approximation)

$$s(\mu) = \sum_{k=0}^{K-1} \int_{t^k}^{t^{k+1}} \ell(u(x, t; \mu)) dt \approx \sum_{k=0}^{K-1} \frac{\Delta t}{2} (\ell(u^k(\mu)) + \ell(u^{k+1}(\mu))). \quad (11)$$

Clearly, with the well-conditions (i.e., symmetric positive definiteness) of the FE mass and stiffness matrices as well as of the initial values, the linear system (10) possesses a unique solution.

The RB approximation shall be built upon our reference finite element approximation, and the RB error will thus be evaluated with respect to $u^k(\mu) \in Y$. Clearly, our methods must remain computationally efficient and stable as $\mathcal{N} \rightarrow \infty$.

Finally, note that our linear and bilinear forms are independent of time – the system is thus linear time-invariant (LTI) [12]. We shall point out that one application which satisfies this property is the dental implant problem [8, 46].

2.3 Dealing with unknown loading

In many dynamical systems, generally, the applied force to excite the system (e.g., $g(t^k)$ in (10) or $g(t)$ in (3)) is not known in advance and thus we cannot solve (9) for $u^{k+1}(\mu)$. In such situations, fortunately, we may appeal to the LTI property to justify an impulse approach as described now [12]. We note from the Duhamel's principle that the solution of any LTI system can be written as the convolution of the impulse response with the control input: for any control input $g_{\text{any}}(t^k)$, we can obtain its corresponding solution $u_{\text{any}}^k(\mu)$, $1 \leq k \leq K$ from

$$u_{\text{any}}^k(\mu) = \sum_{j=1}^k u_{\text{unit}}^{k-j+1}(\mu) g_{\text{any}}(t^j), \quad 1 \leq k \leq K, \quad (12)$$

where $u_{\text{unit}}^k(\mu)$ is the solution of (9) for a unit impulse control input $g_{\text{unit}}(t^k) = \delta_{1k}$, $1 \leq k \leq K$ (δ is the Kronecker delta symbol). Therefore, it is sufficient to build the RB basis functions for the problem based on this impulse response.

3 Reduced basis approximation

Two key properties of the RB methodology will be recalled as follows. First, our attention is restricted to a *smooth* and *low-dimensional* manifold instead of the very high-dimensional FE space. Namely, the field variable $u^k(\mu)$, $1 \leq k \leq K$ does not belong to the very high-dimensional FE space; rather it resides, or “evolves” on a much lower dimensional manifold which is induced by the parametric dependence over the parameter domain [12]. Therefore, by restricting our attention to this manifold, we can adequately

[‡]In order to start the procedure (9), $u^1(\mu)$ is computed as on page 491 of [44].

approximate the field variable by a space of dimension $N \ll \mathcal{N}$. Second, the parametric setting of the PDE (3) enables to split the computational procedure into two stages: an extensive/expensive *Offline* stage performed once to prepare all necessary data for numerous input-output calculations in the *Online* stage afterwards. Details of these computations will be explained in subsequent sections.

3.1 Approximation

We introduce the set of samples $S_* = \{\mu_1 \in \mathcal{D}, \mu_2 \in \mathcal{D}, \dots, \mu_N \in \mathcal{D}\}, 1 \leq N \leq N_{\max}$, and associated nested Lagrangian RB spaces $Y_N = \text{span}\{\zeta_n, 1 \leq n \leq N\}, 1 \leq N \leq N_{\max}$, where $\zeta_n \in Y_N, 1 \leq n \leq N_{\max}$ are mutually $(\cdot, \cdot)_Y$ -orthogonal RB basis functions. The sets S_* and Y_N shall be constructed correspondingly by the standard and goal-oriented POD–Greedy algorithms described in Section 3.2 afterwards.

Our reduced basis approximation $u_N^k(\mu)$ to $u^k(\mu)$ is then obtained by a standard Galerkin projection: given $\mu \in \mathcal{D}$, we now look for $u_N^k(\mu) \in Y_N$ that satisfies

$$\mathcal{A}(u_N^{k+1}(\mu), v; \mu) = \mathcal{F}(v), \quad \forall v \in Y_N, \mu \in \mathcal{D}, 1 \leq k \leq K-1, \quad (13)$$

where

$$\begin{cases} \mathcal{A}(u_N^{k+1}(\mu), v; \mu) &= \frac{1}{\Delta t^2} m(u_N^{k+1}(\mu), v; \mu) + \frac{1}{2\Delta t} c(u_N^{k+1}(\mu), v; \mu) + \frac{1}{4} a(u_N^{k+1}(\mu), v; \mu), \\ \mathcal{F}(v) &= -\frac{1}{\Delta t^2} m(u_N^{k-1}(\mu), v; \mu) + \frac{1}{2\Delta t} c(u_N^{k-1}(\mu), v; \mu) - \frac{1}{4} a(u_N^{k-1}(\mu), v; \mu) \\ &\quad + \frac{2}{\Delta t^2} m(u_N^k(\mu), v; \mu) - \frac{1}{2} a(u_N^k(\mu), v; \mu) + g^{eq}(t^k) f(v; \mu), \end{cases} \quad (14)$$

with the initial conditions: $u_N^0(\mu) = 0, \frac{\partial u_N^0(\mu)}{\partial t} = 0$; we then evaluate the output estimate, $s_N(\mu)$, from

$$s_N(\mu) = \sum_{k=0}^{K-1} \int_{t^k}^{t^{k+1}} \ell(u_N(x, t; \mu)) dt \approx \sum_{k=0}^{K-1} \frac{\Delta t}{2} (\ell(u_N^k(\mu)) + \ell(u_N^{k+1}(\mu))). \quad (15)$$

3.2 Goal-oriented POD–Greedy sampling procedure

3.2.1 The proper orthogonal decomposition

We aim to generate an optimal (in the mean square error sense) basis set $\{\zeta_m\}_{m=1}^M$ from any given set of $M_{\max}(\geq M)$ snapshots $\{\xi_k\}_{k=1}^{M_{\max}}$. To do this, let $V_M = \text{span}\{v_1, \dots, v_M\} \subset \text{span}\{\xi_1, \dots, \xi_{M_{\max}}\}$ be an arbitrary space of dimension M . We assume that the basis $\{v_1, \dots, v_M\}$ is orthonormal such that $(v_n, v_m) = \delta_{nm}, 1 \leq n, m \leq M$ ((\cdot, \cdot) denotes an appropriate inner product and δ_{nm} is the Kronecker delta symbol). The POD space, $W_M = \text{span}\{\zeta_1, \dots, \zeta_M\}$ is defined as

$$W_M = \arg \min_{V_M \subset \text{span}\{\xi_1, \dots, \xi_{M_{\max}}\}} \left(\frac{1}{M_{\max}} \sum_{k=1}^{M_{\max}} \inf_{\alpha^k \in \mathbb{R}^M} \left\| \xi_k - \sum_{m=1}^M \alpha_m^k v_m \right\|^2 \right). \quad (16)$$

In essence, the POD space W_M which is extracted from the given set of snapshots $\{\xi_k\}_{k=1}^{M_{\max}}$ is the space that best approximate this given set of snapshots and can be written as $W_M = \text{POD}(\{\xi_1, \dots, \xi_{M_{\max}}\}, M)$. We can construct this POD space by using the method of snapshots[§] which is presented concisely in the Appendix of [49].

(T1a)	Set $Y_N^{\text{go}} = 0$	Set $Y_N^{\text{st}} = 0$
(T1b)	Set $\mu_*^{\text{go}} = \mu_0$	Set $\mu_*^{\text{st}} = \mu_0$
(T1c)	While $N \leq N_{\max}^{\text{go}}$	While $N \leq N_{\max}^{\text{st}}$
(T1d)	$\mathcal{W}^{\text{go}} = \{e_{\text{proj}}^{\text{go}}(\mu_*^{\text{go}}, t^k), 0 \leq k \leq K\};$	$\mathcal{W}^{\text{st}} = \{e_{\text{proj}}^{\text{st}}(\mu_*^{\text{st}}, t^k), 0 \leq k \leq K\};$
(T1e)	$Y_{N+M}^{\text{go}} \leftarrow Y_N^{\text{go}} \oplus \text{POD}(\mathcal{W}^{\text{go}}, M);$	$Y_{N+M}^{\text{st}} \leftarrow Y_N^{\text{st}} \oplus \text{POD}(\mathcal{W}^{\text{st}}, M);$
(T1f)	$N \leftarrow N + M;$	$N \leftarrow N + M;$
(T1g)	Find \tilde{N} s.t. $\forall \mu \in \Xi_n^{\text{st}} \subset \Xi_{n+1}^{\text{st}} (\subset S_*^{\text{st}}),$ $\eta_T \leq \left \frac{\Delta_s(\mu)}{s(\mu) - s_N^{\text{go}}(\mu)} \right \leq 2 - \eta_T;$	$\mu_*^{\text{st}} = \arg \max_{\mu \in \Xi_{\text{train}}} \left\{ \frac{\Delta_u(\mu)}{\sqrt{\sum_{k=1}^K \ u_N^{\text{st}}(\mu, t^k)\ _Y^2}} \right\};$ $S_*^{\text{st}} \leftarrow S_*^{\text{st}} \cup \{\mu_*^{\text{st}}\};$
(T1h)	$\mu_*^{\text{go}} = \arg \max_{\mu \in \Xi_{\text{train}}} \left\{ \left \frac{\Delta_s(\mu)}{s_N^{\text{st}}(\mu)} \right \right\};$	end.
(T1i)	$S_*^{\text{go}} \leftarrow S_*^{\text{go}} \cup \{\mu_*^{\text{go}}\};$	
(T1j)	end.	
(T1k)	$\Delta_s(\mu) = s_N^{\text{st}}(\mu) - s_N^{\text{go}}(\mu)$	$\Delta_u(\mu) = \sqrt{\sum_{k=1}^K \ \mathcal{R}^{\text{st}}(v; \mu, t^k)\ _Y^2},$

Table 1: (Left) The proposed goal-oriented POD–Greedy sampling algorithm and (Right) the standard POD–Greedy sampling algorithm. (The terms $\Delta_u(\mu)$ and $\Delta_s(\mu)$ are printed in boldface to highlight the difference between the two algorithms.)

3.2.2 Goal-oriented POD–Greedy algorithm

Algorithm 1: The “cross-validation” process.	
INPUT: Ξ_n^{st}, N	
OUTPUT: $\Xi_n^{\text{st}}, \tilde{N}_n$	
1: while true do	
2: Given $\Xi_n^{\text{st}},$	
3: Compute \tilde{N}_n from this Ξ_n^{st} ; (i.e., call the Algorithm 2 below)	
4: Create $\Xi_{n+1}^{\text{st}};$	
5: Check (♣) with \tilde{N}_n over $\Xi_{n+1}^{\text{st}}:$	
6: if (♣) holds $\forall \mu \in \Xi_{n+1}^{\text{st}}$ then	▷ if (♣) holds
7: Get $\{\tilde{N}_n, \Xi_n^{\text{st}}\};$	
8: Exit while loop;	
9: else	▷ if (♣) is violated
10: $\Xi_n^{\text{st}} \leftarrow \Xi_{n+1}^{\text{st}};$	
11: end if	
12: end while	

[§]Some books which investigate thoroughly this POD subject can be found in, for instance, [47, 48].

Algorithm 2: Function to compute \tilde{N} based on input Ξ^{st} and N .

INPUT: Ξ^{st}, N
OUTPUT: \tilde{N}

```

1: Start with  $\tilde{N} = 2N$ ;
2: while true do
3:   Check  $(\clubsuit)$  with  $\tilde{N}$  over  $\Xi^{\text{st}}$ :
4:   if any  $\mu \in \Xi^{\text{st}}$  violates  $(\clubsuit)$  then ▷ if  $(\clubsuit)$  is violated
5:     if  $\tilde{N} < N_{\max}^{\text{st}}$  then
6:        $\tilde{N} \leftarrow \tilde{N} + 1$ ;
7:     else
8:       Run the standard POD–Greedy algorithm to increase  $N_{\max}^{\text{st}}$ :  $N_{\max}^{\text{st}} \leftarrow N_{\max}^{\text{st}} + 1$ ;
9:     end if
10:  end if
11:  if  $(\clubsuit)$  holds  $\forall \mu \in \Xi^{\text{st}}$  then ▷ if  $(\clubsuit)$  holds
12:    Get  $\tilde{N}$ ;
13:    Exit while loop;
14:  end if
15: end while

```

We now discuss the POD–Greedy algorithms [41, 8, 50, 51] to construct the nested sets S_* and Y_N of interest. Let Ξ_{train} be a finite set of the parameters in \mathcal{D} ($\Xi_{\text{train}} \subset \mathcal{D}$); and S_* denote the set of greedily selected parameters in Ξ_{train} . Initialize $S_* = \{\mu_0\}$, where μ_0 is an arbitrarily chosen parameter. Let $e_{\text{proj}}(\mu, t^k) = u^k(\mu) - \text{proj}_{Y_N} u^k(\mu)$, where $\text{proj}_{Y_N} u^k(\mu)$ is the Y_N -orthogonal projection of $u^k(\mu)$ into the Y_N space. Our proposed goal-oriented and the standard algorithms are presented simultaneously in Table 1. Note that the superscript “st” denotes the standard and “go” denotes the goal-oriented algorithms, respectively.

- a) Let us first describe the standard POD–Greedy algorithm in the right column of Table 1. In particular, at each Greedy iteration, one first solves (9) to obtain the “true” FE solution $\{u^k(\mu_*^{\text{st}}), 0 \leq k \leq K\}$; then computes the projection error to form the snapshots set \mathcal{W}^{st} in (T1d). Next, one does snapshots-POD analysis [49] on this set to build/extract M “new” basis functions and adds them to the “current” RB basis functions set Y_N^{st} (T1e). The RB dimension N is also updated correspondingly in (T1f). Now, based on these newly updated bases Y_N^{st} , the error indicator $\frac{\Delta_u(\mu)}{\sqrt{\sum_{k=1}^K \|u_N^{\text{st}}(\mu, t^k)\|_Y^2}}$ will be computed exhaustively for all $\mu \in \Xi_{\text{train}}$ to look for the worst sampling point μ_*^{st} and add it to the set S_*^{st} (T1g). This whole procedure is iterated until it satisfies some stopping criteria. Note in the computation of $\Delta_u(\mu)$ (T1k), the term $\|\mathcal{R}^{\text{st}}(v; \mu, t^k)\|_{Y'}$ is the dual norm of the associated residual of equation (13), namely,

$$\mathcal{R}^{\text{st}}(v; \mu, t^k) = \mathcal{F}(v) - \mathcal{A}(u_N^{k+1}(\mu), v; \mu), \quad \forall v \in Y_N, \mu \in \mathcal{D}, 1 \leq k \leq K-1. \quad (17)$$

Detailed computation of the term $\|\mathcal{R}^{\text{st}}(v; \mu, t^k)\|_{Y'}$ can be found in, for instance, [8, 7, 12]. In essence, the term $\frac{\Delta_u(\mu)}{\sqrt{\sum_{k=1}^K \|u_N^{\text{st}}(\mu, t^k)\|_Y^2}}$ is the ratio of the dual norm of the residual to the norm of the RB solution. Thus, this term is roughly considered as an error indicator for the error in the solution (or field variable). This is one special choice of many ones and is usually used in the current standard POD–Greedy algorithm [41, 17, 8].

- b) We now consider the goal-oriented algorithm in the left column of Table 1. All the main steps of this algorithm are exactly the same as that of the standard algorithm, except that the error indicator is now defined as $\left| \frac{\Delta_s(\mu)}{s_N^{\text{st}}(\mu)} \right|$ in (T1h). Essentially, this term is an asymptotic error approximation for the

true output error (i.e., $s(\mu) - s_N(\mu) \approx s_N^{\text{st}}(\mu) - s_N(\mu)$). Thus, the main idea of this goal-oriented algorithm is that the Greedy iterations now try to minimize the output error indicator rather than the solution error indicator as in the standard algorithm above. From another viewpoint, we can think of this goal-oriented algorithm as a special version of the standard algorithm using a relative output error approximation as an error indicator.

- c) In the goal-oriented algorithm, the computation of $\Delta_s(\mu)$ requires a good approximation $s_{\tilde{N}}^{\text{st}}(\mu)$ for the FE output $s(\mu)$. To cope with this situation, we use the asymptotic output error approximation which makes use of an enriched RB output computation. Specifically, we propose to implement the standard algorithm *in advance* to obtain enriched RB spaces $Y_{N_{\max}}^{\text{st}}$ which are ready for the computation of $s_{\tilde{N}}^{\text{st}}(\mu)$ ($\tilde{N} \geq N$) in the goal-oriented algorithm *afterwards*. Hence, here $s_{\tilde{N}}^{\text{st}}(\mu)$ implies the enriched RB output computed by standard algorithm (“st”) using \tilde{N} basis functions.

The use of asymptotic output error approximation $\Delta_s(\mu)$ is not new in the literature. For example, in the works [32, 33] the authors used heuristically $\tilde{N} = 2N$ as an adequate choice for the output error approximation. In this work, however, we propose a new algorithm called a “cross-validation” process to choose adaptively \tilde{N} for each particular N in the offline stage of the goal-oriented algorithm. Note that these found pairs (N, \tilde{N}) will also be used in the error approximation in the online stage later. The “cross-validation” process is presented in line (T1g) and is detailed in Algorithm 1; however, we postpone its explanation until point f) below.

- d) The main idea of the “cross-validation” process is to find a sufficient \tilde{N} (for an N under consideration) such that the effectivity $\left| \frac{s_{\tilde{N}}^{\text{st}}(\mu) - s_N^{\text{go}}(\mu)}{s(\mu) - s_N^{\text{go}}(\mu)} \right|$ satisfies: $\forall \mu \in \Xi^{\text{try}}$,

$$\eta_T \leq \left| \frac{s_{\tilde{N}}^{\text{st}}(\mu) - s_N^{\text{go}}(\mu)}{s(\mu) - s_N^{\text{go}}(\mu)} \right| \leq 2 - \eta_T, \quad (\clubsuit)$$

where η_T is a user-prescribed effectivity (say, 0.8 or 0.9), and Ξ^{try} is an arbitrary set of parameters ($s_N^{\text{go}}(\mu)$ is a goal-oriented RB output using N basis functions). Because the FE output $s(\mu)$ appears in (\clubsuit) , we think of using $\Xi^{\text{try}} \subset S_*^{\text{st}}$ as all FE solutions/outputs are available for all $\mu \in S_*^{\text{st}}$ (recall that the standard algorithm was *already* implemented). Hence, we will use the notation $\Xi^{\text{st}} (\equiv \Xi^{\text{try}}) \subset S_*^{\text{st}}$ to reflect this idea.

- e) We first explain the Algorithm 2 in detail. Essentially, algorithm 2 is an iteration process to find a proper \tilde{N} which satisfies (\clubsuit) for a given sample set Ξ^{st} and a given N . The iteration procedure starts with $\tilde{N} = 2N$; and \tilde{N} will be increased if (\clubsuit) is violated by any $\mu \in \Xi^{\text{st}}$. The procedure will stop when (\clubsuit) holds true $\forall \mu \in \Xi^{\text{st}}$. Within Algorithm 2, if \tilde{N} exceeds N_{\max}^{st} the standard algorithm will be called and implemented to increase N_{\max}^{st} accordingly. Note that the standard algorithm will continue to run from the previous N_{\max}^{st} ; so in summary, it is deemed to run “once” but continuously in different stages when necessary.
- f) Let us now describe the “cross-validation” process in Algorithm 1. In fact, finding the proper size of Ξ^{st} is not a trivial task: fixing $\Xi^{\text{st}} = S_*^{\text{st}}$ is not an efficient way, and we also don’t want to use one more parameter to tune this setting (the only one parameter for the GO algorithm is η_T). We thus propose an adaptive strategy to choose $|\Xi^{\text{st}}|$ as follows (Algorithm 1). Suppose that at the Greedy iteration N with a given set $\Xi_n^{\text{st}} \subset S_*^{\text{st}}$, we can find the corresponding \tilde{N} thanks to Algorithm 2 above. We will request further that the currently found \tilde{N} also needs to verify (\clubsuit) over Ξ_{n+1}^{st} , where $|\Xi_{n+1}^{\text{st}}| = |\Xi_n^{\text{st}}| + \Delta n_{\text{sample}}$, and Δn_{sample} is a number of next sample points taken from the set S_*^{st} . Otherwise, Ξ_{n+1}^{st} is assigned to Ξ_n^{st} (i.e., Ξ_n^{st} is enriched now) and the procedure is repeated until the found \tilde{N} satisfies (\clubsuit) over both sets Ξ_n^{st} and Ξ_{n+1}^{st} ($\subset S_*^{\text{st}}$). By this way, we can start the sampling procedure with fairly small $|\Xi_n^{\text{st}}|$ and let it “evolve” automatically without the necessity of any control or adjustment. Following the same process, Ξ_n^{st} and Ξ_{n+1}^{st} will also be used to find \tilde{N} at the next Greedy iteration $N + 1$; and they will be enriched appropriately when necessary.

For example, the standard algorithm is implemented *first* and *once* for quite large N_{\max}^{st} , say, $N_{\max}^{\text{st}} = 200$ and hence $|S_*^{\text{st}}| = 200$ (using $M = 1$). Then for an arbitrary quantity of interest, the goal-oriented algorithm will be implemented accordingly. Consider the “cross-validation” process at the first Greedy iteration $N^{\text{go}} = 1$, we can choose $\Xi_1^{\text{st}} \subset \Xi_2^{\text{st}} \subset S_*^{\text{st}}$ such that $|\Xi_1^{\text{st}}| = 10$ and $|\Xi_2^{\text{st}}| = 20$ first sample points of S_*^{st} , respectively (hence, $\Delta n_{\text{sample}} = 10$). Based on this Ξ_1^{st} set, Algorithm 2 is invoked to find the corresponding \tilde{N}_1 . Next, (\clubsuit) is checked with \tilde{N}_1 over Ξ_2^{st} : if it holds true $\forall \mu \in \Xi_2^{\text{st}}$, then $(N^{\text{go}} = 1, \tilde{N}_1)$ will be the necessarily found pair; the algorithm will quit the “cross-validation” process and continue with step (T1h). Otherwise, the algorithm will enrich Ξ_1^{st} and Ξ_2^{st} such that $|\Xi_1^{\text{st}}| = 20$ and $|\Xi_2^{\text{st}}| = 30$ first sample points from S_*^{st} , and repeat the computational procedure until the right pair $(N^{\text{go}} = 1, \tilde{N})$ is found. Ξ_1^{st} and Ξ_2^{st} are also used to check (\clubsuit) at the next Greedy iteration in a completely similar manner.

- g) Finally, we close this subsection with one remark on the possible value range of η_T . In fact, we cannot choose η_T to be too high, i.e., too close to 1. The reason is that the convergence of the GO algorithm depends on the convergence of the standard one; and generally the standard algorithm will stall/flat with some $N \geq N_{\max}^{\text{st}}$ (i.e., its error indicator and RB true error cannot decrease further as it reaches machine accuracy $\approx 10^{-8}$). If η_T is too close to 1, say 0.95, the cross-validation process will break down (infinite loop in Algorithm 1) since it cannot find the suitable \tilde{N} to satisfy (\clubsuit) over Ξ_n^{st} and Ξ_{n+1}^{st} ; and there is no way to handle this situation. Therefore, it is practical to choose a modest η_T , and in fact we can do that easily based on the convergence history of the standard algorithm which is implemented *in advance*. Indeed, through two numerical experiments in Section 4 later, $\eta_T = 0.8$ is the maximum possible choice and it gives the best performance among all GO η_T algorithms. On the contrary, low η_T (i.e., close to 0) poses no difficulty for the cross-validation process since its corresponding \tilde{N} generally will be smaller than that of higher η_T , and hence low η_T is “safer” than high η_T regarding \tilde{N} .

3.3 Error estimations

3.3.1 True errors

We use the true errors to validate the performance of the standard and goal-oriented algorithms in the online computation stage. The relative true errors by the two algorithms for the solutions are defined as

$$e_u^{\text{st}}(\mu) = \sqrt{\frac{\sum_{k=1}^K \|u^k(\mu) - u_N^{\text{st},k}(\mu)\|_Y^2}{\sum_{k=1}^K \|u^k(\mu)\|_Y^2}}, \quad e_u^{\text{go}}(\mu) = \sqrt{\frac{\sum_{k=1}^K \|u^k(\mu) - u_N^{\text{go},k}(\mu)\|_Y^2}{\sum_{k=1}^K \|u^k(\mu)\|_Y^2}}, \quad (18)$$

and for the outputs

$$e_s^{\text{st}}(\mu) = \left| \frac{s(\mu) - s_N^{\text{st}}(\mu)}{s(\mu)} \right| \approx \left| \frac{\sum_{k=0}^{K-1} \frac{\Delta t}{2} \left(\ell(u^k(\mu) - u_N^{\text{st},k}(\mu)) + \ell(u^{k+1}(\mu) - u_N^{\text{st},k+1}(\mu)) \right)}{\sum_{k=0}^{K-1} \frac{\Delta t}{2} (\ell(u^k(\mu)) + \ell(u^{k+1}(\mu)))} \right|, \quad (19a)$$

$$e_s^{\text{go}}(\mu) = \left| \frac{s(\mu) - s_N^{\text{go}}(\mu)}{s(\mu)} \right| \approx \left| \frac{\sum_{k=0}^{K-1} \frac{\Delta t}{2} \left(\ell(u^k(\mu) - u_N^{\text{go},k}(\mu)) + \ell(u^{k+1}(\mu) - u_N^{\text{go},k+1}(\mu)) \right)}{\sum_{k=0}^{K-1} \frac{\Delta t}{2} (\ell(u^k(\mu)) + \ell(u^{k+1}(\mu)))} \right|, \quad (19b)$$

respectively.

In the above expressions, $u^k(\mu)$, $u_N^{\text{st},k}(\mu)$ and $u_N^{\text{go},k}(\mu)$ are the FE, standard RB and goal-oriented RB solutions; $s(\mu)$, $s_N^{\text{st}}(\mu)$ and $s_N^{\text{go}}(\mu)$ are the FE, standard RB and goal-oriented RB outputs, respectively.

3.3.2 Output error approximation

The true errors are good for validation purposes but are not of practical uses in the online stage, where one requires fast and countless online calculations. In this work, we propose to use $\Delta_s(\mu)$ as an error estimation for the output in the online computation stage (in short, $\Delta_s(\mu)$ is an error estimation in both offline and online stages). Of course this error approximation is not a rigorous upper error bound such as the *a posteriori* error bounds [7, 15, 17]; however, there are several good reasons for using it in practice. First, the time-marching error bounds for the wave equation so far were shown to be ineffective and pessimistic due to the instability of the wave equation: exponential growing with respect to time [19, 52, 53]. (We also note that the space-time error bounds, although very promising, are still not yet derived in the literature.) Second, this error approximation converges asymptotically to the true error (thanks to \tilde{N} chosen effectively by the proposed “cross-validation” process), and thus can approximate relatively the accuracy of the RB outputs for various choices of μ . Thirdly – most important, its computational cost is cheap: only $O(N^3) + O(\tilde{N}^3)$ as described in the next section ($N, \tilde{N} \ll \mathcal{N}$). The output error approximation $\Delta_s(\mu)$ in (T1k) and its associated effectivity are defined as follows

$$\Delta_s^{\text{go}}(\mu) = s_N^{\text{st}}(\mu) - s_N^{\text{go}}(\mu), \quad \eta_s^{\text{go}}(\mu) = \left| \frac{\Delta_s^{\text{go}}(\mu)}{s(\mu) - s_N^{\text{go}}(\mu)} \right|. \quad (20)$$

Note that to compare the performance of output error approximation of the standard versus the goal-oriented algorithms, here we also define the output error approximation and its associated effectivity for the standard algorithm as

$$\Delta_s^{\text{st}}(\mu) = s_N^{\text{st}}(\mu) - s_N^{\text{st}}(\mu), \quad \eta_s^{\text{st}}(\mu) = \left| \frac{\Delta_s^{\text{st}}(\mu)}{s(\mu) - s_N^{\text{st}}(\mu)} \right|, \quad (21)$$

where the superscript “st” implies the standard algorithm. The superscript “go” is thus added in (20) to imply the goal-oriented algorithm, respectively.

3.4 Offline-online computational procedure

In this section, we develop offline-online computational procedures in order to fully exploit the dimension reduction of the problem [12, 8, 15]. We note that both algorithms (standard and goal-oriented) have the same offline-online computational procedures, they are only different in the ways to build the sets S_* and Y_N via Greedy iterations. We first express $u_N^k(\mu)$ as:

$$u_N^k(\mu) = \sum_{n=1}^N u_{Nn}^k(\mu) \zeta_n, \quad \forall \zeta_n \in Y_N. \quad (22)$$

We then choose a test function $v = \zeta_n, 1 \leq n \leq N$ for the RB equation (13). It then follows that $\underline{u}_N^k(\mu) = [u_{N1}^k(\mu) \ u_{N2}^k(\mu) \ \cdots \ u_{NN}^k(\mu)]^T \in \mathbb{R}^N$ satisfies

$$\begin{aligned} & \left(\frac{1}{\Delta t^2} \mathbf{M}_N(\mu) + \frac{1}{2\Delta t} \mathbf{C}_N(\mu) + \frac{1}{4} \mathbf{A}_N(\mu) \right) \underline{u}_N^{k+1}(\mu) \\ &= \left(-\frac{1}{\Delta t^2} \mathbf{M}_N(\mu) + \frac{1}{2\Delta t} \mathbf{C}_N(\mu) - \frac{1}{4} \mathbf{A}_N(\mu) \right) \underline{u}_N^{k-1}(\mu) \\ &+ \left(\frac{2}{\Delta t^2} \mathbf{M}_N(\mu) - \frac{1}{2} \mathbf{A}_N(\mu) \right) \underline{u}_N^k(\mu) + g^{eq}(t^k) \mathbf{F}_N(\mu), \quad 1 \leq k \leq K-1. \end{aligned} \quad (23)$$

The initial condition is treated similar to the treatment in (9) and (13). Here, $\mathbf{C}_N(\mu)$, $\mathbf{A}_N(\mu)$, $\mathbf{M}_N(\mu) \in \mathbb{R}^{N \times N}$ are symmetric positive definite matrices[¶] with entries $\mathbf{C}_{N\,i,j}(\mu) = c(\zeta_i, \zeta_j; \mu)$, $\mathbf{A}_{N\,i,j}(\mu) = a(\zeta_i, \zeta_j; \mu)$, $\mathbf{M}_{N\,i,j}(\mu) = m(\zeta_i, \zeta_j; \mu)$, $1 \leq i, j \leq N$ and $\mathbf{F}_N \in \mathbb{R}^N$ is the RB load vector with entries $\mathbf{F}_{N\,i} = f(\zeta_i)$, $1 \leq i \leq N$, respectively.

The RB output is then computed from

$$s_N(\mu) = \sum_{k=0}^{K-1} \frac{\Delta t}{2} \mathbf{L}_N^T (\underline{u}_N^k(\mu) + \underline{u}_N^{k+1}(\mu)). \quad (24)$$

Invoking the affine parameter dependence (8), we obtain

$$\mathbf{M}_{N\,i,j}(\mu) = m(\zeta_i, \zeta_j; \mu) = \sum_{q=1}^{Q_m} \Theta_m^q(\mu) m^q(\zeta_i, \zeta_j), \quad (25a)$$

$$\mathbf{C}_{N\,i,j}(\mu) = c(\zeta_i, \zeta_j; \mu) = \sum_{q=1}^{Q_c} \Theta_c^q(\mu) c^q(\zeta_i, \zeta_j), \quad (25b)$$

$$\mathbf{A}_{N\,i,j}(\mu) = a(\zeta_i, \zeta_j; \mu) = \sum_{q=1}^{Q_a} \Theta_a^q(\mu) a^q(\zeta_i, \zeta_j), \quad (25c)$$

$$\mathbf{F}_{N\,i}(\mu) = f(\zeta_i; \mu) = \sum_{q=1}^{Q_f} \Theta_f^q(\mu) f^q(\zeta_i), \quad (25d)$$

which can be written as

$$\begin{aligned} \mathbf{M}_{N\,i,j}(\mu) &= \sum_{q=1}^{Q_m} \Theta_m^q(\mu) \mathbf{M}_{N\,i,j}^q, & \mathbf{C}_{N\,i,j}(\mu) &= \sum_{q=1}^{Q_c} \Theta_c^q(\mu) \mathbf{C}_{N\,i,j}^q, \\ \mathbf{A}_{N\,i,j}(\mu) &= \sum_{q=1}^{Q_a} \Theta_a^q(\mu) \mathbf{A}_{N\,i,j}^q, & \mathbf{F}_{N\,i}(\mu) &= \sum_{q=1}^{Q_f} \Theta_f^q(\mu) \mathbf{F}_{N\,i}^q, \end{aligned} \quad (26)$$

where the *parameter independent* quantities \mathbf{M}_N^q , \mathbf{C}_N^q , $\mathbf{A}_N^q \in \mathbb{R}^{N \times N}$, and $\mathbf{F}_N^q \in \mathbb{R}^N$ are given by

$$\mathbf{M}_{N\,i,j}^q = m^q(\zeta_i, \zeta_j), \quad 1 \leq i, j \leq N_{\max}, \quad 1 \leq q \leq Q_m, \quad (27a)$$

$$\mathbf{C}_{N\,i,j}^q = c^q(\zeta_i, \zeta_j), \quad 1 \leq i, j \leq N_{\max}, \quad 1 \leq q \leq Q_c, \quad (27b)$$

$$\mathbf{A}_{N\,i,j}^q = a^q(\zeta_i, \zeta_j), \quad 1 \leq i, j \leq N_{\max}, \quad 1 \leq q \leq Q_a, \quad (27c)$$

$$\mathbf{F}_{N\,i}^q = f^q(\zeta_i), \quad 1 \leq i \leq N_{\max}, \quad 1 \leq q \leq Q_f, \quad (27d)$$

respectively.

The offline-online computational procedure is now described as follows. In the offline stage – performed only once, we first implement the standard POD–Greedy algorithm [8]: we solve to find $Y_N^{\text{st}} = \{\zeta_n^{\text{st}}, 1 \leq n \leq N_{\max}\}$; then compute and store the μ -independent quantities in (27) for the estimation of the RB solution and output^{||}. Once these RB solution and output are available, we can now implement the goal-oriented POD–Greedy algorithm. We consider each goal-oriented POD–Greedy

[¶]The proof of this property can be found in, for instance, Proposition 5.1, page 136 of [54]. Thanks to this property, the stability of our proposed RB scheme will be guaranteed as a consequence.

^{||}There are still several terms related to the computation of the dual norm of the residual $\|\mathcal{R}^{\text{st}}(v; \mu, t^k)\|_{Y'}$, we do not show them here for simplicity. Detailed implementation of the standard POD–Greedy algorithm can be referred to, for instance, [8, 17, 15].

iteration (Table 1) in more details. We first need to solve (9) for the “true” FE solutions; then compute the projection error in step (T1d) and solve the POD/eigenvalue problem in step (T1e). In addition, we have to compute $O(N^2Q)$ \mathcal{N} -inner products in (27) (we denote $Q = Q_m + Q_c + Q_a$). By approximating $s(\mu)$ via the enriched approximation $s_N^{\text{st}}(\mu)$ in (T1k) through the standard algorithm, we can now do an exhaustive yet cheap search over Ξ_{train} to look for the optimal μ in each Greedy iteration. In summary, the offline stage of the goal-oriented algorithm also includes the offline stage of the standard algorithm, and therefore it is more expensive than that of the standard algorithm. (We again emphasize that the standard algorithm is implemented only *once*, and then the goal-oriented algorithm can be implemented *many* times corresponding with various output functionals, respectively.)

The online stage of the goal-oriented algorithm is very similar to that of the standard algorithm [8]. In the online stage – performed many times, for each new parameter μ – we first assemble the RB matrices in (25), this requires $O(N^2Q)$ operations. We then solve the RB governing equation (23), the operation counts are $O(N^3 + KN^2)$ as the RB matrices are generally full. Finally, we evaluate the displacement output $s_N(\mu)$ from (24) at the cost of $O(KN)$. For the error approximation (i.e., $\Delta_s(\mu) = s_N^{\text{st}}(\mu) - s_N^{\text{go}}(\mu)$), there is nothing more than computing one more output $s_N^{\text{st}}(\mu)$ and then performing the associated subtraction; the cost is $O(\tilde{N}^3)$. Therefore, as required in the real-time context, the online complexity to compute the output and its associated error approximation are $O(N^3) + O(\tilde{N}^3)$ – independent of \mathcal{N} ; and since $N, \tilde{N} \ll \mathcal{N}$ we can expect significant computational saving in the online stage relative to the classical FE approach.

4 Numerical examples

In this section, we will verify both POD–Greedy algorithms by investigating a simple 2D linear elastodynamic problem and a 3D dental implant model problem in the time domain. The details are described in the following.

4.1 A 2D linear elastodynamic problem

4.1.1 Finite element model and approximation

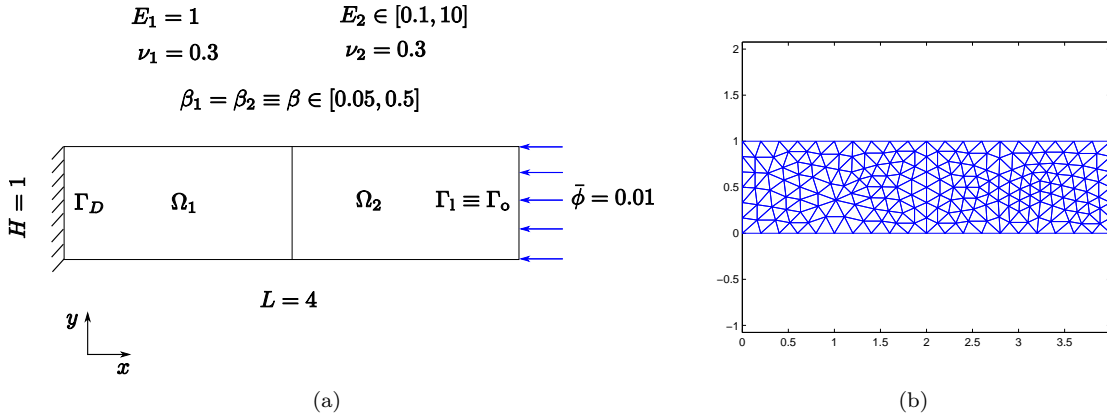


Figure 1: (a) The 2D model with its dimensionless parameters, and (b) its FE reference mesh.

We consider a 2D plane strain model problem as in Fig.1(a). It is assumed that the model problem is scaled (or non-dimensionalized) from a real problem in practice and hence all the terms are dimensionless. The details of the non-dimensionalization is briefly discussed in Appendix A. The length and height of the model are $L = 4$ and $H = 1$, respectively. The model is composed of 2

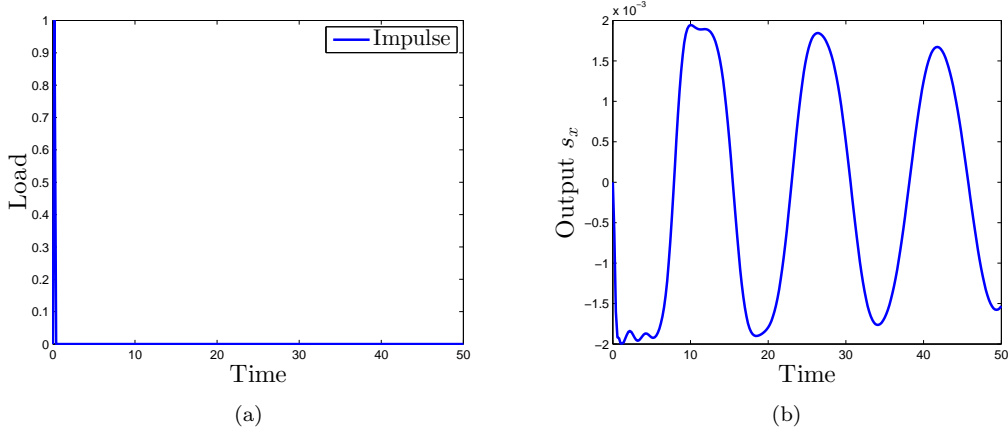


Figure 2: (a) Time history of impulse loading, and (b) corresponding FEM output displacement in x -direction versus time with $\mu_{\text{test}} = (1, 0.1)$.

subdomains Ω_1 and Ω_2 with two different materials: Young's moduli $E_1 = 1$ and $E_2 \in [0.1, 10]$ and Poisson ratios $\nu_1 = \nu_2 = 0.3$, respectively. We assume Rayleigh damping for the model where the mass-proportional damping coefficients $\alpha_1 = \alpha_2 = 0$, and the stiffness-proportional damping coefficients $\beta_1 = \beta_2 \equiv \beta \in [0.05, 0.5]$ such that $\mathbf{C}_i = \beta_i \mathbf{A}_i$, $1 \leq i \leq 2$, where \mathbf{C}_i and \mathbf{A}_i are the FEM damping and stiffness matrices of each subdomain, respectively. Isotropic and homogenous material behavior is assumed for the model. We also note that the material mass densities will vanish from the weak form of the PDE due to the non-dimensionalisation (eq.(34), Appendix A). A surface traction which is opposite to the x -direction with the magnitude $\bar{\phi} = 0.01$ is then applied to the right edge of the model (Γ_1) as shown in Fig.1(a). As mentioned in Section 2.3, it is sufficient to perform all calculations for the impulse loading case since the solutions associated with other loading cases can be inferred by Duhamel's convolution principle. Time history of the impulse loading is also shown in Fig.2(a). Homogeneous Dirichlet boundary condition is enforced on the left edge (Γ_D), while zero initial conditions (i.e., zero displacement and velocity) are applied on the model. The output of interest is defined as the time integral of the average x -displacement along the right edge ($\Gamma_o \equiv \Gamma_1$) of the model.

Specifically, the explicit forms of all linear and bilinear forms associated with the 2D model problem are defined as follows

$$m(w, v) = \sum_{r=1}^2 \sum_i \int_{\Omega_r} w_i v_i, \quad (28a)$$

$$a(w, v; \mu) = \sum_{i,j,k,l} \int_{\Omega_1} \frac{\partial v_i}{\partial x_j} C_{ijkl}^1 \frac{\partial w_k}{\partial x_l} + \mu_1 \sum_{i,j,k,l} \int_{\Omega_2} \frac{\partial v_i}{\partial x_j} C_{ijkl}^2 \frac{\partial w_k}{\partial x_l}, \quad (28b)$$

$$c(w, v; \mu) = \mu_2 \sum_{i,j,k,l} \int_{\Omega_1} \frac{\partial v_i}{\partial x_j} C_{ijkl}^1 \frac{\partial w_k}{\partial x_l} + \mu_2 \mu_1 \sum_{i,j,k,l} \int_{\Omega_2} \frac{\partial v_i}{\partial x_j} C_{ijkl}^2 \frac{\partial w_k}{\partial x_l}, \quad (28c)$$

$$f(v) = \sum_i \int_{\Gamma_1} v_i \bar{\phi}_i, \quad (28d)$$

$$\ell(v) = \frac{1}{|\Gamma_o|} \int_{\Gamma_o} v_1, \quad (28e)$$

for all $w, v \in Y$, $\mu \in \mathcal{D}$. Here, the parameter $\mu = (\mu_1, \mu_2) \equiv (E_2, \beta)$, where E_2 is the Young's modulus of the domain Ω_2 and β is the stiffness-proportional damping coefficient of both domains Ω_1, Ω_2 . C_{ijkl}^r is the constitutive elasticity tensor for isotropic materials and it is expressed in terms of the Young's modulus E and Poisson's ratio ν of each region $\Omega_r, 1 \leq r \leq 2$, respectively. It is obvious from (8) and (28) that the smooth functions $\Theta_a^1(\mu) = 1$, $\Theta_a^2(\mu) = \mu_1$; $\Theta_c^1(\mu) = \mu_2$, $\Theta_c^2(\mu) = \mu_1\mu_2$ depend on μ — but the bilinear forms $a^1(w, v) = c^1(w, v) = \sum_{i,j,k,l} \int_{\Omega_1} \frac{\partial v_i}{\partial x_j} C_{ijkl}^1 \frac{\partial w_k}{\partial x_l}$, and $a^2(w, v) = c^2(w, v) = \sum_{i,j,k,l} \int_{\Omega_2} \frac{\partial v_i}{\partial x_j} C_{ijkl}^2 \frac{\partial w_k}{\partial x_l}$ do not depend on μ .

The finite element mesh consist of 215 nodes and 376 linear triangular elements as shown in Fig.1(b). The FE space to approximate the 2D elastodynamic problem is of dimension $\mathcal{N} = 416$. For time integration, $T = 50$, $\Delta t = 0.2$, $K = \frac{T}{\Delta t} = 250$. The parameter $\mu = (E_2, \beta) \in \mathcal{D}$, where the parameter domain $\mathcal{D} \equiv [0.1, 10] \times [0.05, 0.5] \subset \mathbb{R}^{P=2}$. The $\|\cdot\|_Y$ used in this work is defined as $\|w\|_Y^2 = a(w, w; \bar{\mu}) + m(w, w; \bar{\mu})$, where $\bar{\mu} = (1, 0.1)$; $Q_a = 2$, $Q_c = 2$. The entire work is performed using the software Matlab R2012b. We finally show in Fig.2(b) the “unit” FEM output displacement (i.e., under the unit impulse load) in the x -direction versus time at μ_{test} , where $\mu_{\text{test}} = \bar{\mu} = (1, 0.1)$.

4.1.2 Numerical results

The impulse loading case For this 2D model problem, we aim to investigate the behavior of the goal-oriented algorithms with various choices of η_T under the impulse loading regime. To start, a training sample set Ξ_{train} is created by a equidistant distribution over \mathcal{D} with $n_{\text{train}} (= 30 \times 30) = 900$ samples. Note that we use $M = 1$ and $N_{\text{max}}^{\text{go}} = 60$ (as in Table 1) to terminate the iteration procedures. In the remaining sections, beside the standard and goal-oriented algorithms (η_T), we also show the results of the goal-oriented algorithm ($\tilde{N} = 2N$) for comparison purpose.

- a) First, we implement the standard POD–Greedy algorithm (i.e., the right column of Table 1) [8]. The results are presented in Fig.3: Fig.3(a) shows the maximum error indicator $\Delta_u^{\text{max,rel}} = \max_{\mu \in \Xi_{\text{train}}} \left\{ \frac{\Delta_u(\mu)}{\sqrt{\sum_{k=1}^K \|u_N^{\text{st}}(\mu, t^k)\|_Y^2}} \right\}$ as a function of N ; and Fig.3(b) shows the distribution of the sample set S_*^{st} , respectively. As observe from Fig.3(a), the max error indicator $\Delta_u^{\text{max,rel}}$ could not decrease further when $N^{\text{st}} \geq 140$, hence we set $N_{\text{max}}^{\text{st}} = 150$ for the implementation of the GO algorithms afterwards. (Of course, $N_{\text{max}}^{\text{st}}$ is still able to be enriched automatically whenever $\tilde{N} > N_{\text{max}}^{\text{st}}$ as described in Algorithm 2 of the “cross-validation” process, Section 3.2.2.) For comparison purpose, the results associated with the first $N_{\text{max}}^{\text{st}} = 60$ basis functions were plotted using different markers in Fig.3, respectively.
- b) Once the sets S_*^{st} and Y_N^{st} are available, the term $s_N^{\text{st}}(\mu)$ (of $\Delta_s(\mu)$ in (T1k)) is now computable; and hence, it is possible to implement the goal-oriented POD–Greedy algorithms (i.e., the left column of Table 1). For comparison, we show the results of the goal-oriented algorithms using $(\tilde{N} = 2N)$, $(\tilde{N}, \eta_T = 0.8)$, $(\tilde{N}, \eta_T = 0.5)$ and $(\tilde{N}, \eta_T = 0.1)$ in Fig.4, respectively. Note that there is no “cross-validation” process for the first algorithm as we just fix $\tilde{N} = 2N$ for the computation of $s_N^{\text{st}}(\mu)$ in $\Delta_s(\mu)$. The latter algorithms use the proposed “cross-validation” process to choose adaptively \tilde{N} with the settings: $|\Xi_1^{\text{st}}| = 10$ and $\Delta n_{\text{sample}} = 10$ corresponding with $\eta_T = 0.8, 0.5$ and 0.1 , respectively. Fig.4(a) presents the maximum error indicators $\Delta_s^{\text{max,rel}} = \max_{\mu \in \Xi_{\text{train}}} \left\{ \left| \frac{\Delta_s(\mu)}{s_N^{\text{st}}(\mu)} \right| \right\}$ as a function of N ; and Fig.4(b) shows the distribution of the sample sets S_*^{go} . The results of the “cross-validation” processes $(\tilde{N}, \eta_T = 0.8)$, $(\tilde{N}, \eta_T = 0.5)$ and $(\tilde{N}, \eta_T = 0.1)$ are also presented in Fig.5. In particular, Fig.5(a) presents all the \tilde{N} found adaptively for each N ; Fig.5(b) shows the size of Ξ_n^{st} as a function of N ; Fig.5(c) and Fig.5(d) illustrate the maximum and minimum effectivities in (♣) over Ξ_n^{st} , respectively. Fig.5 confirms the behavior of these GO algorithms: as η_T increases close to 1, the range $[\eta_T, 2 - \eta_T]$ shrinks around 1 (i.e., better test effectivities), hence the algorithm would need larger \tilde{N} to hold (♣) true.

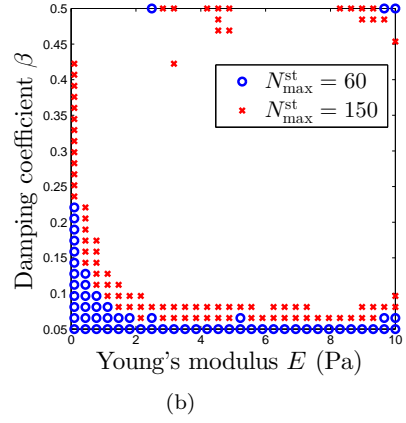
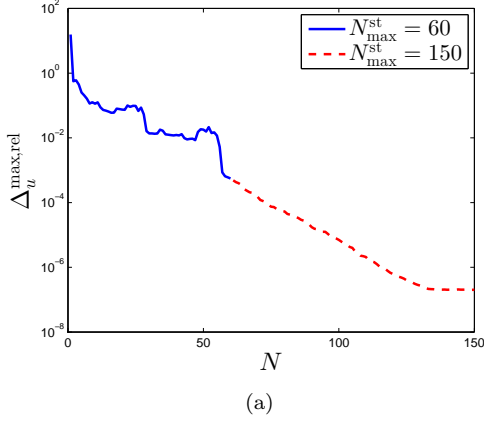


Figure 3: (a) Maximum of error indicator $\Delta_u^{\max, \text{rel}}$ over Ξ_{train} as a function of N , and (b) distribution of sampling points by the standard POD-Greedy algorithm ($N_{\text{max}}^{\text{st}} = 150$). Different markers were used for the first $N_{\text{max}}^{\text{st}} = 60$ basis functions.

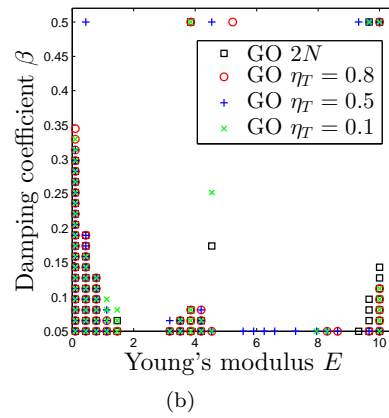
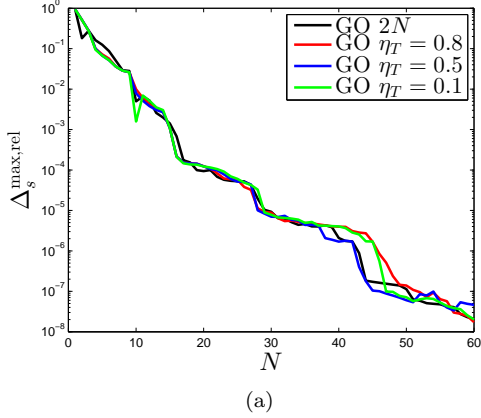


Figure 4: (a) Maximum of error indicator $\Delta_s^{\max, \text{rel}}$ over Ξ_{train} as a function of N , and (b) distribution of sampling points by the goal-oriented POD-Greedy algorithms ($\tilde{N} = 2N$), ($\tilde{N}, \eta_T = 0.8$), ($\tilde{N}, \eta_T = 0.5$) and ($\tilde{N}, \eta_T = 0.1$).

c) In order to evaluate the performance of the algorithms, we create a test sample set $\Xi_{\text{test}} \subset \mathcal{D}$ which is a coarse subset of \mathcal{D} ; then compute and compare mutually their RB true errors. This set has $n_{\text{test}} (= 10 \times 10) = 100$ sample points distributed equidistantly. We show, as functions of N : $e_u^{\max} = \max_{\mu \in \Xi_{\text{test}}} e_u(\mu)$ (defined in (18)) and $e_s^{\max} = \max_{\mu \in \Xi_{\text{test}}} e_s(\mu)$ (defined in (19)) in Fig.6, respectively. Fig.6(a) shows quite similar true solution errors by the algorithms; while from Fig.6(b), the goal-oriented algorithms give true output errors smaller than that computed by the standard algorithm. The differences in the RB true output errors by the standard versus GO algorithms can be up to 10 times. Therefore, we can conclude that the GO algorithms (with cross-validation processes) give RB true output errors smaller than that provided by the standard algorithm (for the same N basis functions under consideration).

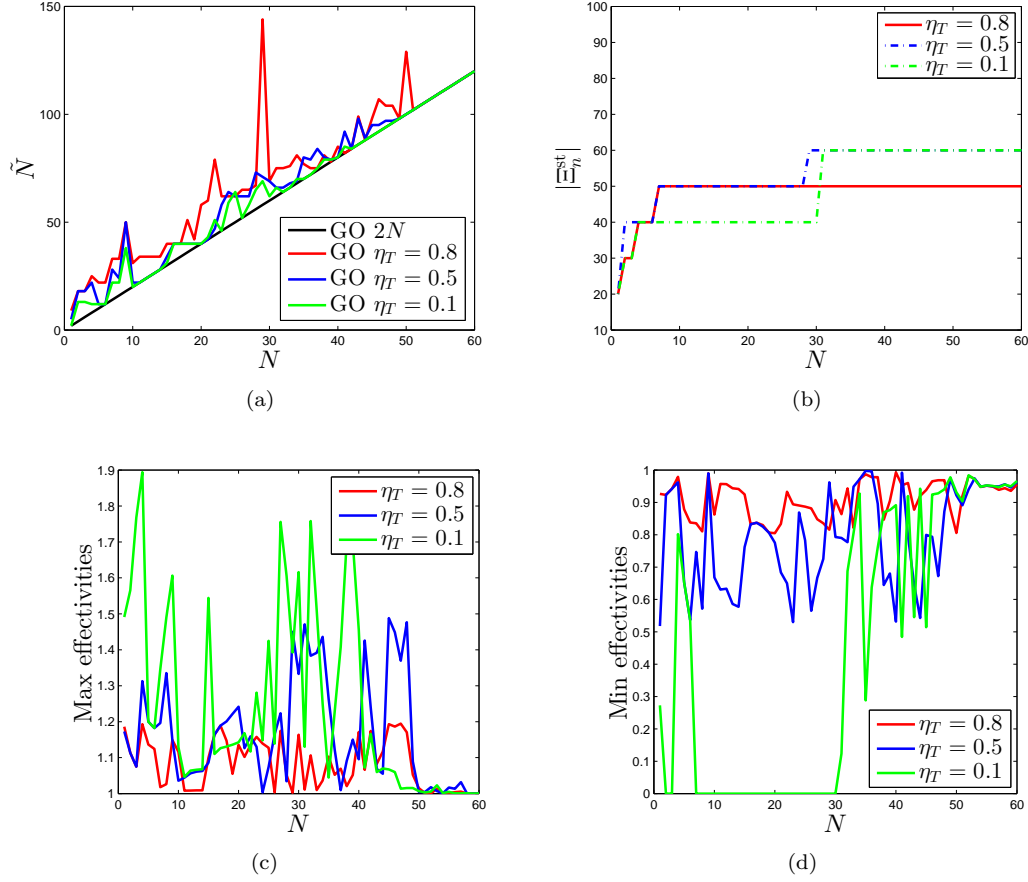


Figure 5: (a) \tilde{N} , (b) the size $|\Xi_n^{\text{st}}|$, (c) max, and (d) min effectivities $\left| \frac{\Delta_s(\mu)}{s(\mu) - s_N^{\text{GO}}(\mu)} \right|$ in \clubsuit as functions of N of the GO algorithms $(\tilde{N}, \eta_T = 0.8)$, $(\tilde{N}, \eta_T = 0.5)$ and $(\tilde{N}, \eta_T = 0.1)$.

d) Lastly, we evaluate the performance of the sampling algorithms via their (practical) output error approximations. For comparison, the output error approximation (21) by the standard algorithm with $(\tilde{N}, \eta_T = 0.8)$ is used. (Namely, the pairs (N, \tilde{N}) in (21) is taken following and after implementing the GO $(\tilde{N}, \eta_T = 0.8)$ algorithm**.) In particular, Fig.7(b) shows the graphs of $\tilde{\Delta}_s^{\text{max,rel}} = \max_{\mu \in \Xi_{\text{test}}} \left| \frac{\Delta_s(\mu)}{s(\mu)} \right|$ (defined in (20), (21)) as functions of N by the algorithms: standard $(\tilde{N}, \eta_T = 0.8)$, GO $(\tilde{N} = 2N)$, GO $(\tilde{N}, \eta_T = 0.8)$, GO $(\tilde{N}, \eta_T = 0.5)$ and GO $(\tilde{N}, \eta_T = 0.1)$, respectively. Fig.7(a) is just a repetition of Fig.6(b) – which shows the max relative RB true errors in order to compare with the max output error approximations in Fig.7(b). The associated maximum/minimum effectivities of this error approximation (i.e., $\eta_s^{\text{max}} = \max_{\mu \in \Xi_{\text{test}}} \eta_s(\mu)$, $\eta_s^{\text{min}} = \min_{\mu \in \Xi_{\text{test}}} \eta_s(\mu)$, defined in (20), (21)) are also shown in Fig.7(c) and Fig.7(d), respectively. As observed from Fig.7(a) and Fig.7(b), the GO algorithms are superior to the standard algorithm in both RB true error and output error approximation. Regarding effectivities, Fig.7(c) and Fig.7(d) show that the GO $(\tilde{N}, \eta_T = 0.8)$ algorithm gives the best effectivities among all the algorithms. These results

**The aim of this special test is that for the same online computational effort $O(N^3) + O(\tilde{N}^3)$, we want to compare the error approximations and associated effectivities by the standard vs. the GO $(\tilde{N}, \eta_T = 0.8)$ algorithms.

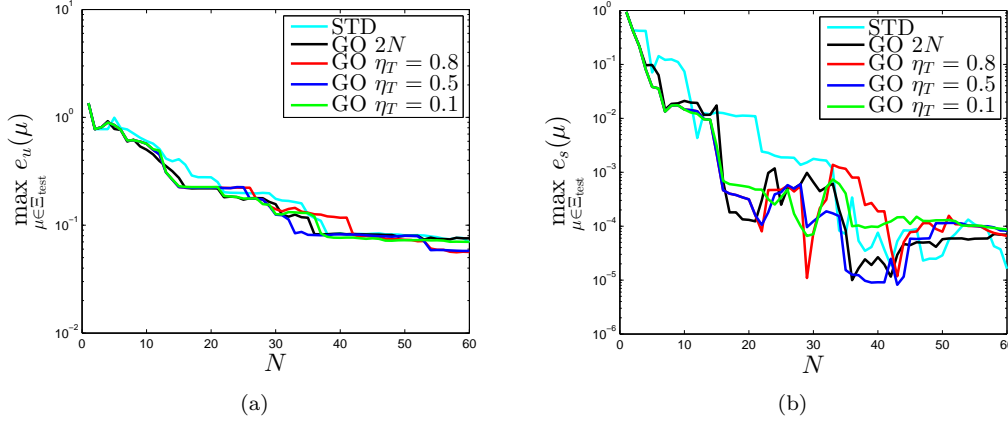


Figure 6: Comparison of maximum (relative) RB true errors by standard and all goal-oriented POD-Greedy algorithms over Ξ_{test} : (a) solution and (b) output.

confirm the good performance of the “cross-validation” process: the GO ($\tilde{N}, \eta_T = 0.8$) algorithm not only decreases the RB true error compared with the standard algorithm, but it also improves significantly the output effectivity estimation compared with other GO algorithms (especially the GO ($\tilde{N} = 2N$) one).

4.2 A 3D dental implant model problem

4.2.1 Finite element model and approximation

We now consider a simplified 3D dental implant-bone model in Fig.8(a). The geometry of the simplified dental implant-bone model is constructed by using the software SolidWorks 2010. The physical domain Ω consists of five regions: the outermost cortical bone Ω_1 , the cancellous bone Ω_2 , the interfacial tissue Ω_3 , the dental implant Ω_4 and the stainless steel screw Ω_5 . The 3D simplified model is then meshed using the software ABAQUS/CAE version 6.10-1 (Fig.8(b)). A dynamic force opposite to the x -direction is then applied to a prescribed area on the body of the screw as shown in Fig.9(a). As mentioned in Sections 2.3 and 4.1.1, all computations and simulations will be performed for the impulse loading case thanks to the Duhamel’s convolution principle (12). Fig.9(b) shows the time history of both loading cases used in this work: the impulse load and an “arbitrary” load. The Dirichlet boundary condition ($\partial\Omega^D$) is specified in the bottom-half of the simplified model as illustrated in Fig.9(a); while zero initial conditions (i.e., zero displacement and velocity) are applied on the model. The output of interest is defined as the average x -displacement responses of a prescribed area on the head of the screw (Fig.9(a)). The finite element mesh consists of 9479 nodes and 50388 four-node tetrahedral solid elements. The coinciding nodes of the contact surfaces between different regions (the regions $\Omega_1, \Omega_2, \Omega_3, \Omega_4, \Omega_5$) are assumed to be rigidly fixed, i.e. the displacements in the x -, y - and z -directions are all set to be the same for the same coinciding nodes.

We assume that the regions $\Omega_i, 1 \leq i \leq 5$, of the simplified model are homogeneous and isotropic. The material properties: the Young’s moduli, Poisson’s ratios and densities of these regions are presented in Table 2 [55]. As similar to [8], we still use Rayleigh damping with stiffness-proportional damping coefficient $\beta_i, 1 \leq i \leq 5$ (Table 2) such that $\mathbf{C}_i = \beta_i \mathbf{A}_i, 1 \leq i \leq 5$, where \mathbf{C}_i and \mathbf{A}_i are the FEM damping and stiffness matrices of each region, respectively. We also note in Table 2 that (E, β) – the Young’s modulus and Rayleigh damping coefficient associated with the interfacial tissue are our sole parameters.

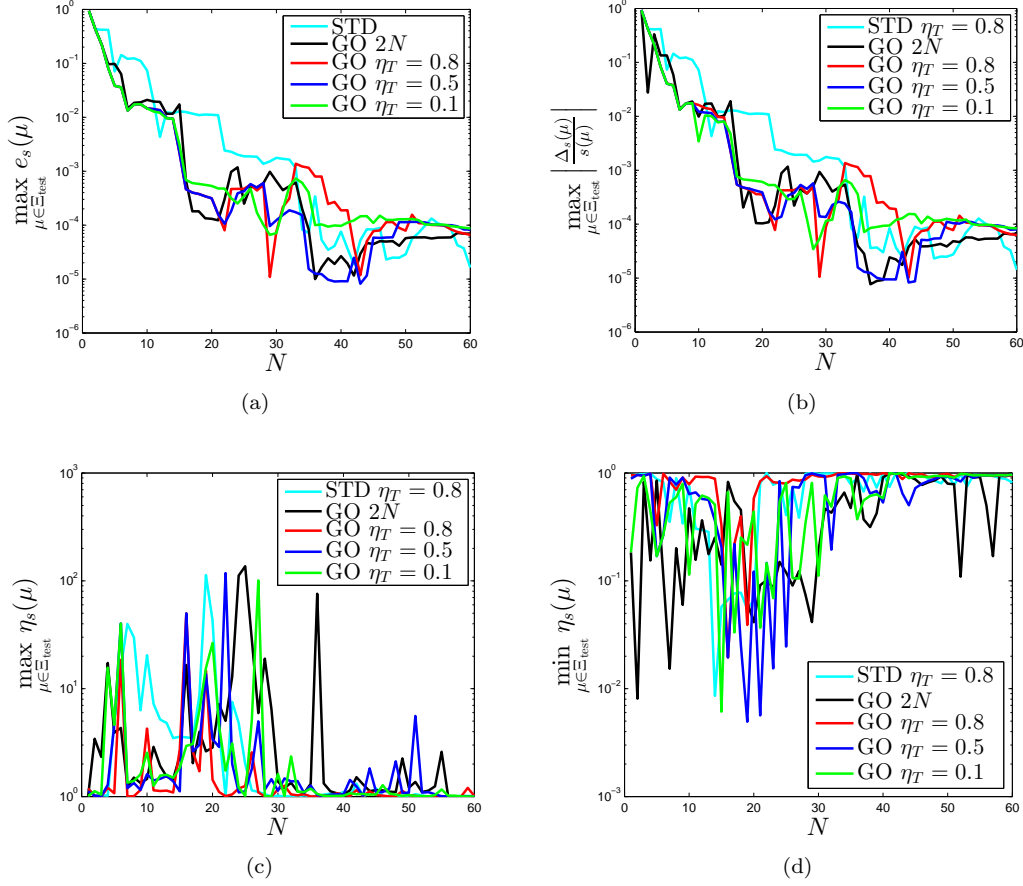


Figure 7: “Impulse load” case: comparison of (a) max relative RB true errors, (b) max relative error approximations, (c) corresponding max effectivities and (d) corresponding min effectivities of all sampling algorithms over Ξ_{test} .

Domain	Layers	E (Pa)	ν	$\rho(\text{g/mm}^3)$	β
Ω_1	Cortical bone	2.3162×10^{10}	0.371	1.8601×10^{-3}	3.38×10^{-6}
Ω_2	Cancellous bone	8.2345×10^8	0.3136	7.1195×10^{-4}	6.76×10^{-6}
Ω_3	Tissue	E	0.3155	1.055×10^{-3}	β
Ω_4	Titan implant	1.05×10^{11}	0.32	4.52×10^{-3}	5.1791×10^{-10}
Ω_5	Stainless steel screw	1.93×10^{11}	0.305	8.027×10^{-3}	2.5685×10^{-8}

Table 2: Material properties of the dental implant-bone structure.

With respect to our particular dental implant problem, the actual integral forms of the linear and bilinear forms are defined as:

$$m(w, v) = \sum_{r=1}^5 \sum_i \int_{\Omega_r} \rho_r w_i v_i, \quad (29a)$$

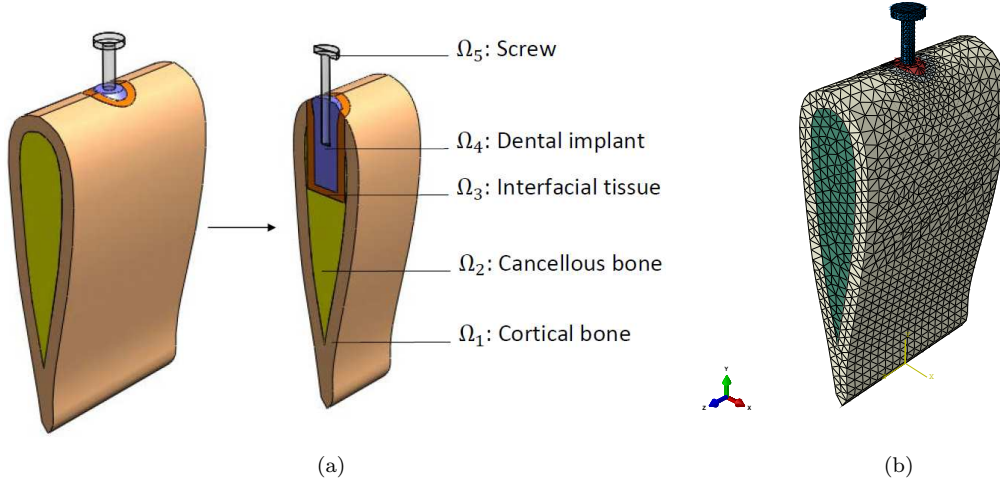


Figure 8: (a) The 3D simplified FEM model with sectional view, and (b) meshing in ABAQUS.

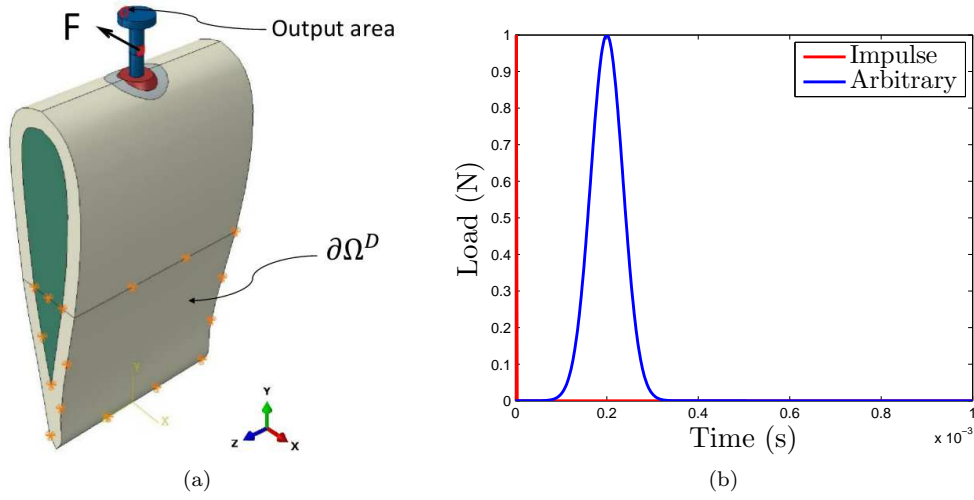


Figure 9: (a) Output area, applied load \mathbf{F} and boundary condition, and (b) time history of an impulse and an “arbitrary” load.

$$a(w, v; \mu) = \sum_{r=1, r \neq 3}^5 \sum_{i,j,k,l} \int_{\Omega_r} \frac{\partial v_i}{\partial x_j} C_{ijkl}^r \frac{\partial w_k}{\partial x_l} + \mu_1 \sum_{i,j,k,l} \int_{\Omega_3} \frac{\partial v_i}{\partial x_j} C_{ijkl}^3 \frac{\partial w_k}{\partial x_l}, \quad (29b)$$

$$c(w, v; \mu) = \sum_{r=1, r \neq 3}^5 \beta_r \sum_{i,j,k,l} \int_{\Omega_r} \frac{\partial v_i}{\partial x_j} C_{ijkl}^r \frac{\partial w_k}{\partial x_l} + \mu_2 \mu_1 \sum_{i,j,k,l} \int_{\Omega_3} \frac{\partial v_i}{\partial x_j} C_{ijkl}^3 \frac{\partial w_k}{\partial x_l}, \quad (29c)$$

$$f(v) = \sum_i \int_{\Gamma_1} v_i \bar{\phi}_i, \quad (29d)$$

$$\ell(v) = \frac{1}{|\Gamma_o|} \int_{\Gamma_o} v_1, \quad (29e)$$

for all $w, v \in Y$, $\mu \in \mathcal{D}$. Here, the parameter $\mu = (\mu_1, \mu_2) \equiv (E, \beta)$ belongs to the region Ω_3 . C_{ijkl}^r is the constitutive elasticity tensor for isotropic materials and it is expressed in terms of the Young's modulus E and Poisson's ratio ν of each region Ω_r , $1 \leq r \leq 5$, respectively. Γ_l is the prescribed loading area (surface traction) and Γ_o is the prescribed output area as shown in Fig.9(a), respectively. It is obvious from (8) and (29) that the smooth functions $\Theta_a^1(\mu) = 1$, $\Theta_a^2(\mu) = \mu_1$; $\Theta_c^1(\mu) = 1$, $\Theta_c^2(\mu) = \mu_1\mu_2$ depend on μ — but the bilinear forms $a^1(w, v) = \sum_{r=1, r \neq 3}^5 \sum_{i,j,k,l} \int_{\Omega_r} \frac{\partial v_i}{\partial x_j} C_{ijkl}^r \frac{\partial w_k}{\partial x_l}$, $a^2(w, v) = c^2(w, v) = \sum_{i,j,k,l} \int_{\Omega_3} \frac{\partial v_i}{\partial x_j} C_{ijkl}^3 \frac{\partial w_k}{\partial x_l}$ and $c^1(w, v) = \sum_{r=1, r \neq 3}^5 \beta_r \sum_{i,j,k,l} \int_{\Omega_r} \frac{\partial v_i}{\partial x_j} C_{ijkl}^r \frac{\partial w_k}{\partial x_l}$ do not depend on μ .

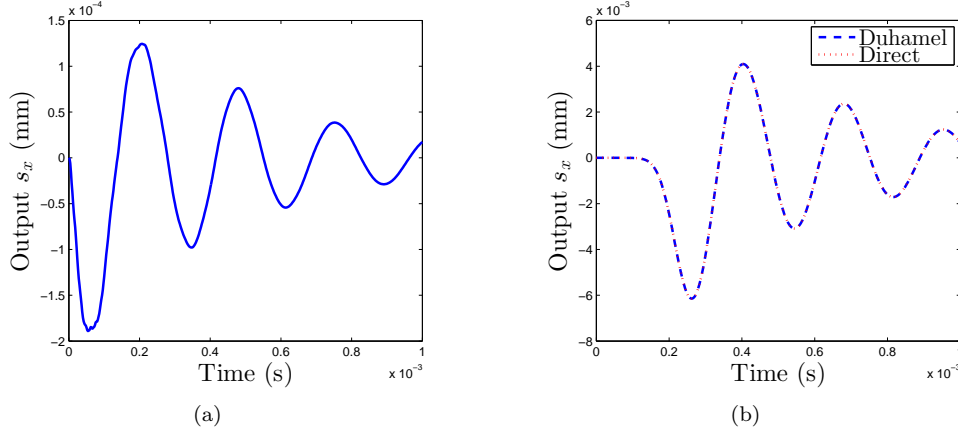


Figure 10: (a) “Unit” FEM output displacement in x –direction versus time, and (b) comparison of “arbitrary” FEM output displacements (under the applied load in Fig.9(b)) computed by Duhamel’s convolution and direct computation, with $\mu_{\text{test}} = (10 \times 10^6 \text{Pa}, 1 \times 10^{-5})$.

The FE space to approximate the 3D dental implant-bone problem is of dimension $\mathcal{N} = 26343$. For time integration, $T = 1 \times 10^{-3} \text{s}$, $\Delta t = 2 \times 10^{-6} \text{s}$, $K = \frac{T}{\Delta t} = 500$. The input parameter $\mu \equiv (E, \beta) \in \mathcal{D}$, where the parameter domain $\mathcal{D} \equiv [1 \times 10^6, 25 \times 10^6] \text{Pa} \times [5 \times 10^{-6}, 5 \times 10^{-5}] \subset \mathbb{R}^{P=2}$. (Note that the range of E of this parameter domain is nearly two times larger than that of [8].) The $\|\cdot\|_Y$ norm used in this work is defined as $\|w\|_Y^2 = a(w, w; \bar{\mu}) + m(w, w; \bar{\mu})$, where $\bar{\mu} = (13 \times 10^6 \text{Pa}, 2.75 \times 10^{-5})$ is the arithmetic average of μ in \mathcal{D} ; $Q_a = 2$, $Q_c = 2$. The entire work is performed using the software Matlab R2012b. Fig.10 presents some FEM results corresponding with the test parameter $\mu_{\text{test}} = (10 \times 10^6 \text{Pa}, 1 \times 10^{-5})$. In particular, Fig.10(a) shows the “unit” FEM output displacement (i.e., under the unit impulse load) in the x –direction versus time at μ_{test} . In addition, Fig.10(b) shows the FEM output displacements versus time under the “arbitrary” load (in Fig.9(b)) by direct FEM computation and by Duhamel’s convolution. It is observed that these two results match perfectly well with each other.

4.2.2 Numerical results

The impulse loading case We now discuss the POD–Greedy algorithms of interest. A training sample set Ξ_{train} is created by a equidistant distribution over \mathcal{D} with $n_{\text{train}} (= 30 \times 30) = 900$ samples. We still use $M = 1$ and $N_{\text{max}}^{\text{go}} = 60$ (Table 1) to terminate the iteration procedures. In the remaining sections, we only show the results of 3 typical algorithms which are the standard, the GO ($\tilde{N}, \eta_T = 0.8$) and the GO ($\tilde{N} = 2N$) algorithms for comparison.

- a) As similar to Section 4.1.2 above, the standard POD–Greedy algorithm is implemented first. The results are presented in Fig.11: Fig.11(a) shows the maximum error indicator $\Delta_u^{\text{max,rel}} =$

$\max_{\mu \in \Xi_{\text{train}}} \left\{ \frac{\Delta_u(\mu)}{\sqrt{\sum_{k=1}^K \|u_N^{\text{st}}(\mu, t^k)\|_Y^2}} \right\}$ as a function of N ; and Fig.11(b) shows the distribution of the sample set S_*^{st} , respectively. $N_{\text{max}}^{\text{st}}$ is set to be 200 since the goal-oriented algorithms may need a quite large $N_{\text{max}}^{\text{st}}$ afterwards. For comparison purpose, the results associated with the first $N_{\text{max}}^{\text{st}} = 60$ basis functions were also plotted using different markers as in Fig.11, respectively.

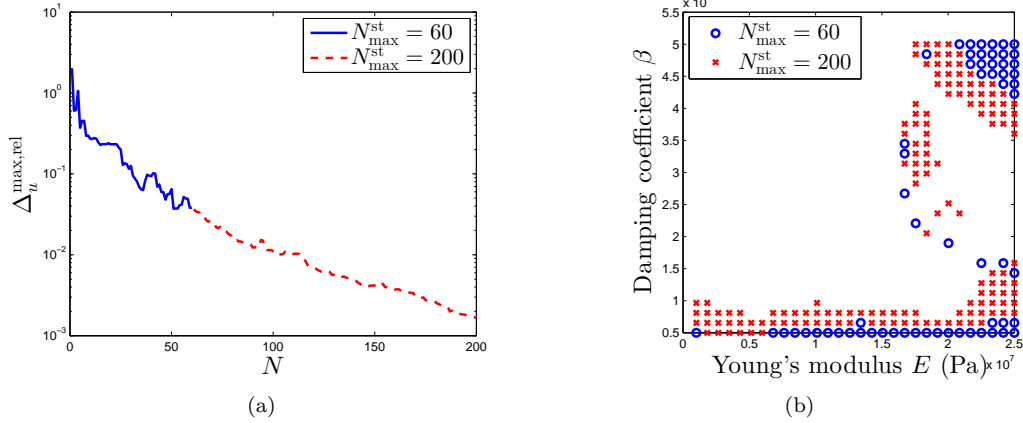


Figure 11: (a) Maximum of error indicator $\Delta_u^{\text{max,rel}}$ over Ξ_{train} as a function of N , and (b) distribution of sampling points by the standard POD-Greedy algorithm ($N_{\text{max}}^{\text{st}} = 200$). Different markers were used for the first $N_{\text{max}}^{\text{st}} = 60$ basis functions.

b) Now, the goal-oriented POD-Greedy algorithms can be implemented since the term $s_N^{\text{st}}(\mu)$ is computable from the availability of the sets S_*^{st} and Y_N^{st} . For comparison, we show the results of the goal-oriented algorithms using $(\tilde{N} = 2N)$ versus $(\tilde{N}, \eta_T = 0.8)$ in Fig.12. The settings $|\Xi_1^{\text{st}}| = 10$ and $\Delta n_{\text{sample}} = 10$ are used for this GO $(\tilde{N}, \eta_T = 0.8)$ algorithm. Fig.12(a) presents the maximum error indicators $\Delta_s^{\text{max,rel}} = \max_{\mu \in \Xi_{\text{train}}} \left\{ \left| \frac{\Delta_s(\mu)}{s_N^{\text{st}}(\mu)} \right| \right\}$ as a function of N ; and Fig.12(b) shows the distribution of the sample sets S_*^{go} , respectively. It is observed from Fig.12(a) that the GO $(\tilde{N}, \eta_T = 0.8)$ algorithm is just a bit better than the GO $(\tilde{N} = 2N)$ one regarding maximum error indicators. However, Fig.12(b) shows that these two algorithms are different from each other by 15/60 (=1/4) sample points. This is an important difference which will be reflected clearly in the effectivity estimation by the two algorithms afterwards. The results of the “cross-validation” process $(\tilde{N}, \eta_T = 0.8)$ are also presented in Fig.13. In particular, Fig.13(a) presents all the \tilde{N} found adaptively for each N ; Fig.13(b) illustrates the maximum and minimum effectivities in (\clubsuit) over Ξ_n^{st} ; and Fig.13(c) shows the size of Ξ_n^{st} as a function of N , respectively.

c) In order to evaluate the performance of the algorithms, we create a test sample set $\Xi_{\text{test}} \subset \mathcal{D}$ which is a coarse subset of \mathcal{D} ; then compute and compare mutually their RB true errors. This set has $n_{\text{test}} (= 10 \times 10) = 100$ sample points distributed equidistantly. We show, as functions of N : $e_u^{\text{max}} = \max_{\mu \in \Xi_{\text{test}}} e_u(\mu)$ (defined in (18)) and $e_s^{\text{max}} = \max_{\mu \in \Xi_{\text{test}}} e_s(\mu)$ (defined in (19)) in Fig.14, respectively. We observe again from Fig.14(b) that the goal-oriented algorithms completely prevail over the the standard algorithm as regards true output errors. The differences in the RB true output errors by the standard versus GO algorithms can be up to 10 times. As expected from Fig.14(b), the GO $(\tilde{N}, \eta_T = 0.8)$ algorithm is a bit better than the GO $(\tilde{N} = 2N)$ algorithm because the former uses more RB basis functions (\tilde{N}) than the latter with the same N (see Fig.13(a)).

d) Finally, we evaluate the performance of the sampling algorithms via their (practical) output

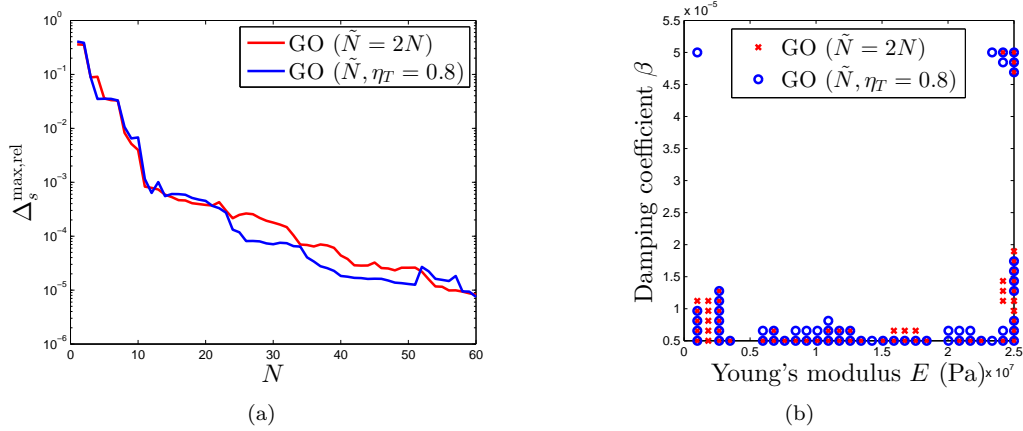


Figure 12: (a) Maximum of error indicator $\Delta_s^{\max, \text{rel}}$ over Ξ_{train} as a function of N , and (b) distribution of sampling points by the goal-oriented POD-Greedy algorithms ($\tilde{N} = 2N$) v.s. ($\tilde{N}, \eta_T = 0.8$).

error approximations. Again, the output error approximation (21) by the standard algorithm with $(\tilde{N}, \eta_T = 0.8)$ is used. (Namely, the pairs (N, \tilde{N}) in (21) is taken following and after implementing the $\text{GO } (\tilde{N}, \eta_T = 0.8)$ algorithm.) Specifically, Fig.15(b) shows the graphs of $\tilde{\Delta}_s^{\max, \text{rel}} = \max_{\mu \in \Xi_{\text{test}}} \left| \frac{\Delta_s(\mu)}{s(\mu)} \right|$ (defined in (20), (21)) as functions of N by 3 algorithms: standard ($\tilde{N}, \eta_T = 0.8$), $\text{GO } (\tilde{N}, \eta_T = 0.8)$ and $\text{GO } (\tilde{N} = 2N)$, respectively. Fig.15(a) is again a repetition of Fig.14(b) – which shows the max relative RB true errors to compare with the max output error approximations in Fig.15(b). The associated maximum/minimum effectivities of this error approximation (i.e., $\eta_s^{\max} = \max_{\mu \in \Xi_{\text{test}}} \eta_s(\mu)$, $\eta_s^{\min} = \min_{\mu \in \Xi_{\text{test}}} \eta_s(\mu)$, defined in (20), (21)) are also shown in Fig.15(c) and Fig.15(d), respectively. Fig.15(a) and Fig.15(b) again show that the GO algorithms give smaller RB true output errors than that provided by the standard algorithm (for the same N basis functions). Fig.15(c) and Fig.15(d) also show much better effectivities of the standard and $\text{GO } (\tilde{N}, \eta_T = 0.8)$ algorithms compared with that of the $\text{GO } (\tilde{N} = 2N)$ one. These results again confirm the good performance of the “cross-validation” process: the $\text{GO } (\tilde{N}, \eta_T = 0.8)$ algorithm decreases the RB true error compared with the standard algorithm, and it also improves significantly the output effectivity estimation compared with the $\text{GO } (\tilde{N} = 2N)$ algorithm.

An arbitrary loading case All the results shown so far are for the unit impulse load, which is of very limited use in practice. Fortunately, as mentioned in Section 2.3, we can use the Duhamel’s principle to convolute all the “unit” RB solutions with an “arbitrary” load to obtain correspondingly “arbitrary” RB solutions/outputs. Here, we also test and present some results of our error approximation $\Delta_s(\mu)$ with the “arbitrary” load defined in Fig.9(b). We note that the error approximation (T1k) is good only for the impulse load since we built the goal-oriented RB basis functions for this load. For more complicated loadings, the error approximation might be much worse since there might be error in the convolution. Similar to paragraph d) of Section 4.2.2 above, Fig.16(a) and Fig.16(b) show the graphs of max relative RB true error and max relative error approximation; while Fig.16(c) and Fig.16(d) show the corresponding max/min effectivities of the 3 algorithms: standard, $\text{GO } (\tilde{N}, \eta_T = 0.8)$ and $\text{GO } (\tilde{N} = 2N)$ over Ξ_{test} , respectively. Again, we obtain quite similar phenomena as in Fig.15: the GO algorithms beat the standard one in RB true error and error approximation; and the $\text{GO } (\tilde{N}, \eta_T = 0.8)$ algorithm provides the best effectivities among all algorithms. The results again prove the preeminence of the proposed “cross-validation” process to choose adaptively and sufficiently \tilde{N} for

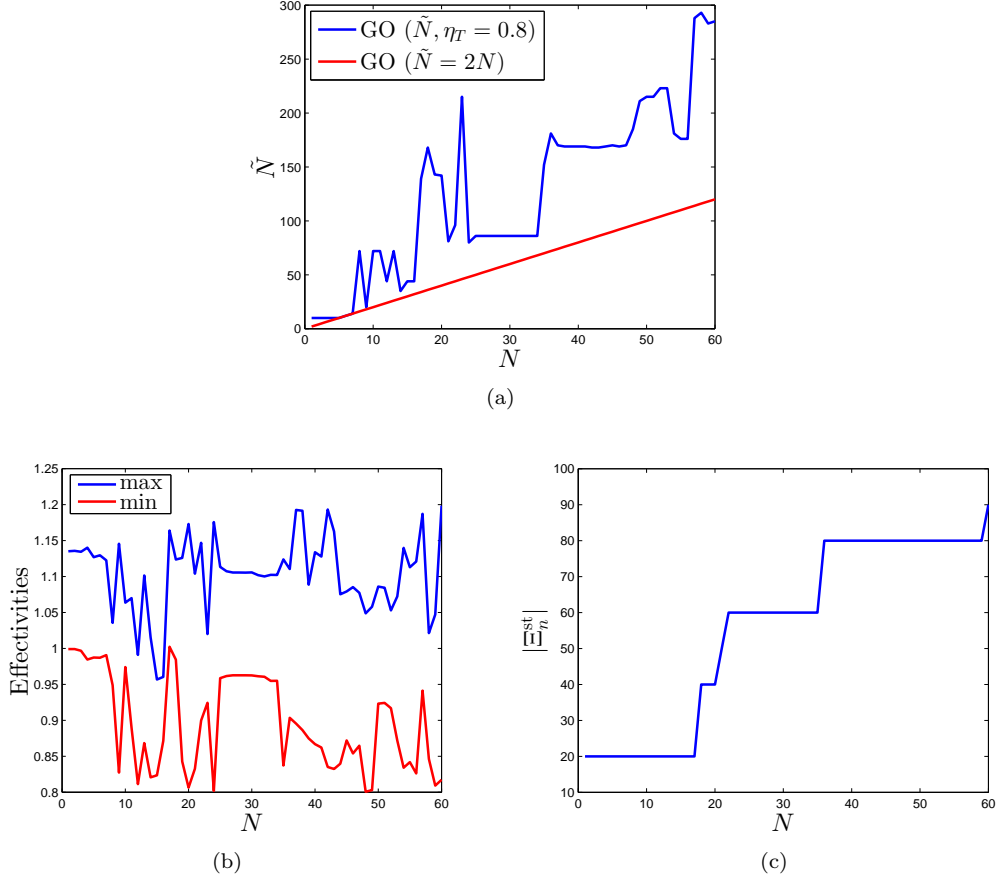


Figure 13: (a) \tilde{N} , (b) max/min effectivities $\left| \frac{\Delta_s(\mu)}{s(\mu) - s_N^{\text{go}}(\mu)} \right|$ in (\clubsuit) , $\forall \mu \in \Xi_n^{\text{st}}$, and (c) the size $|\Xi_n^{\text{st}}|$ as functions of N of the algorithm GO $(\tilde{N}, \eta_T = 0.8)$.

a particular N of interest.

Finally, regarding computational time, all computations were performed on a desktop Intel(R) Core(TM) i7-3930K CPU @3.20GHz 3.20GHz, RAM 32GB, 64-bit Operating System. The computational time for the RB solver ($t_{\text{RB}(\text{online})}$), the CPU-time for the FEM solver by our code (t_{FEM}) and the CPU-time saving factor $\kappa = t_{\text{FEM}}/t_{\text{RB}(\text{online})}$ are listed on Table 3, respectively. We also provide roughly the computational time for the error approximations ($\Delta_s(\mu)$) using both GO algorithms in that Table. We see that the RB solver is approximately $O(1000)$ times and the error approximation is $O(100)$ faster than the FEM solver; and thus it is clear that the RB is very efficient and reliable for solving time-dependent dynamic problems [56, 57, 58, 59, 60].

5 Conclusion

A new goal-oriented POD–Greedy sampling algorithm was proposed. The proposed algorithm cooperates and improves further the standard POD–Greedy algorithm by using the output error approximation rather than the dual norm of residual as error indicator within the Greedy iterations. A cross-validation process is proposed to choose adaptively the dimension of the larger-dimensional auxiliary RB space in

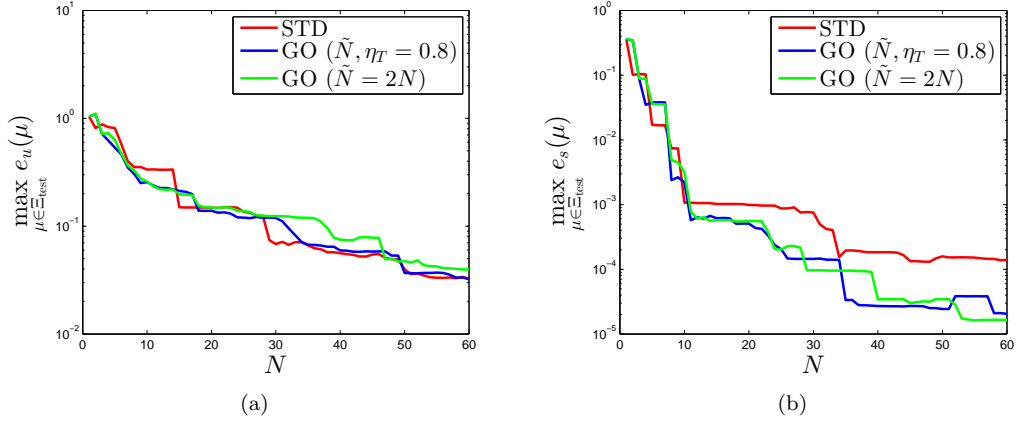


Figure 14: Comparison of maximum (relative) RB true errors by standard and all goal-oriented POD-Greedy algorithms over Ξ_{test} : (a) solution and (b) output.

Table 3: Comparison of the CPU-time for a FEM and RB analysis.

N	$t_{\text{RB(online)}} \text{ (sec)}$	$t_{\text{FEM}} \text{ (sec)}$	$\kappa = t_{\text{FEM}}/t_{\text{RB(online)}}$	$t_{\Delta_s(\mu)} \text{ (sec)}$ ($\tilde{N} = 2N$)	$t_{\Delta_s(\mu)} \text{ (sec)}$ ($\tilde{N}, \eta_T = 0.8$)
10	0.006757	29	4291	0.008329	0.038035
20	0.008391	29	3456	0.014242	0.099046
30	0.011123	29	2607	0.031723	0.049015
40	0.014531	29	1996	0.042440	0.130987
50	0.024381	29	1189	0.061122	0.192829
60	0.031196	29	930	0.075049	0.328012

the output error approximation. The proposed strategy is verified by investigating a 2D linear plane strain problem and a 3D dental implant problem in the time domain. It is demonstrated that this type of error indicator will guide the Greedy iterations to select the parameter samples to optimize the true output error. In comparison with the standard algorithm, we conclude that our proposed algorithm performs better – in terms of output’s accuracy, and quite similar – in terms of solution’s accuracy. The proposed algorithm is applicable to various (regular) output functionals and is thus very suitable within the goal-oriented RB approximation context.

Acknowledgements

We are sincerely grateful for the financial support of the European Research Council Starting Independent Research Grant (ERC Stg grant agreement No. 279578) entitled “Towards real time multiscale simulation of cutting in non-linear materials with applications to surgical simulation and computer guided surgery”. We also thank the anonymous reviewers for many contributive comments which improve the quality of this paper.

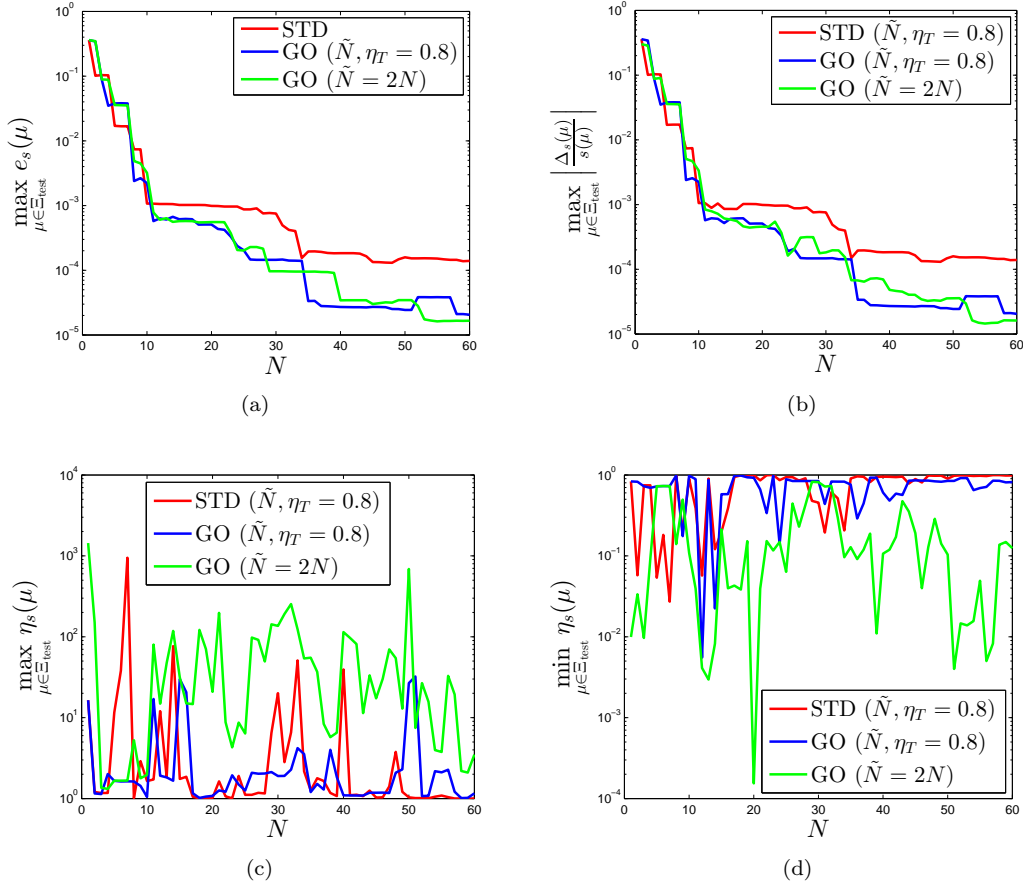


Figure 15: “Impulse load” case: comparison of (a) max relative RB true errors, (b) max relative error approximations, (c) corresponding max effectivities and (d) corresponding min effectivities of all sampling algorithms over Ξ_{test} .

References

- [1] L. Sirovich, “Turbulence and the dynamics of coherent structures, part 1: Coherent structures,” *Quarterly of applied mathematics*, vol. 45, pp. 561–571, 1987.
- [2] P. Kerfriden, P. Gosselet, S. Adhikari, S. Bordas, and J.-C. Passieux, “POD-based model order reduction for the simulation of strong nonlinear evolutions in structures: application to damage propagation,” in *IOP Conference Series: Materials Science and Engineering*, vol. 10, p. 012165, IOP Publishing, 2010.
- [3] P. Kerfriden, J.-C. Passieux, and S. P.-A. Bordas, “Local/global model order reduction strategy for the simulation of quasi-brittle fracture,” *International Journal for Numerical Methods in Engineering*, vol. 89, no. 2, pp. 154–179, 2011.
- [4] P. Kerfriden, P. Gosselet, S. Adhikari, and S. P.-A. Bordas, “Bridging proper orthogonal decomposition methods and augmented Newton–Krylov algorithms: an adaptive model order reduction for highly nonlinear mechanical problems,” *Computer Methods in Applied Mechanics and Engineering*, vol. 200, no. 5, pp. 850–866, 2011.

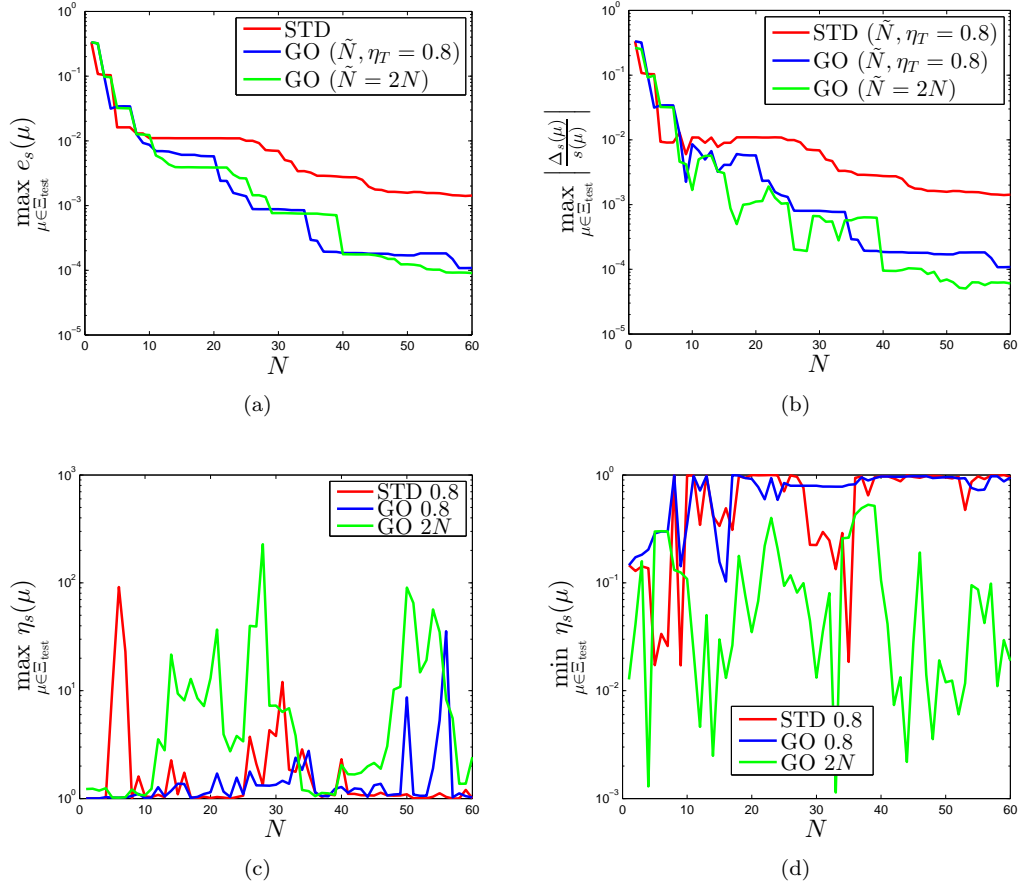


Figure 16: “Arbitrary load” case: comparison of (a) max relative RB true errors, (b) max relative error approximations, (c) corresponding max effectivities and (d) corresponding min effectivities of all sampling algorithms over Ξ_{test} . (Shorter legends were used in Fig.16(c) and Fig.16(d) to avoid affecting the graphs.)

- [5] P. Kerfriden, O. Gouy, T. Rabczuk, and S. P.-A. Bordas, “A partitioned model order reduction approach to rationalise computational expenses in nonlinear fracture mechanics,” *Computer methods in applied mechanics and engineering*, vol. 256, pp. 169–188, 2012.
- [6] P. Kerfriden, J.-J. Ródenas, and S.-A. Bordas, “Certification of projection-based reduced order modelling in computational homogenisation by the constitutive relation error,” *International Journal for Numerical Methods in Engineering*, vol. 97, no. 6, pp. 395–422, 2014.
- [7] G. Rozza, D. B. P. Huynh, and A. T. Patera, “Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations,” *Archives of Computational Methods in Engineering*, vol. 15, no. 3, pp. 229–275, 2008.
- [8] K. C. Hoang, B. C. Khoo, G. R. Liu, N. C. Nguyen, and A. T. Patera, “Rapid identification of material properties of the interface tissue in dental implant systems using reduced basis method,” *Inverse Problems in Science and Engineering*, vol. 21, no. 8, pp. 1310–1334, 2013.

- [9] A. K. Noor and J. M. Peters, “[Reduced basis technique for nonlinear analysis of structures](#),” *AIAA Journal*, vol. 18, no. 4, pp. 455–462, 1980.
- [10] L. Machiels, Y. Maday, I. B. Oliveira, A. T. Patera, and D. V. Rovas, “[Output bounds for reduced-basis approximations of symmetric positive definite eigenvalue problems](#),” *Comptes Rendus de l’Académie des Sciences-Series I-Mathematics*, vol. 331, no. 2, pp. 153–158, 2000.
- [11] K. Veroy, C. Prudhomme, D. V. Rovas, and A. T. Patera, “[A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations](#),” in *Proceedings of the 16th AIAA computational fluid dynamics conference*, vol. 3847, 2003.
- [12] M. A. Grepl and A. T. Patera, “[A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations](#),” *ESAIM: Mathematical Modelling and Numerical Analysis*, vol. 39, no. 01, pp. 157–181, 2005.
- [13] M. A. Grepl, Y. Maday, N. C. Nguyen, and A. T. Patera, “[Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations](#),” *ESAIM: Mathematical Modelling and Numerical Analysis*, vol. 41, no. 03, pp. 575–605, 2007.
- [14] A. Y. K. Tan, “[Reduced Basis Method for 2nd Order Wave Equation: Application to One-Dimensional Seismic Problem](#),” Master’s thesis, Massachusetts Institute of Technology, 2007.
- [15] N.-C. Nguyen, G. Rozza, and A. T. Patera, “[Reduced basis approximation and a posteriori error estimation for the time-dependent viscous Burgers equation](#),” *Calcolo*, vol. 46, no. 3, pp. 157–185, 2009.
- [16] N. Jung, B. Haasdonk, and D. Kroner, “[Reduced basis method for quadratically nonlinear transport equations](#),” *International Journal of Computing Science and Mathematics*, vol. 2, no. 4, pp. 334–353, 2009.
- [17] D. J. Knezevic, N.-C. Nguyen, and A. T. Patera, “[Reduced basis approximation and a posteriori error estimation for the parametrized unsteady Boussinesq equations](#),” *Mathematical Models and Methods in Applied Sciences*, vol. 21, no. 07, pp. 1415–1442, 2011.
- [18] D. B. P. Huynh, D. J. Knezevic, and A. T. Patera, “[A Laplace transform certified reduced basis method; application to the heat equation and wave equation](#),” *Comptes Rendus Mathématique*, vol. 349, no. 7, pp. 401–405, 2011.
- [19] K. C. Hoang, *Reduced Basis Approximation and Inverse Analyses for Dental Implant Problems*. PhD thesis, Singapore–MIT Alliance, National University of Singapore, 2012.
- [20] T. Grätsch and K.-J. Bathe, “[A posteriori error estimation techniques in practical finite element analysis](#),” *Computers & structures*, vol. 83, no. 4, pp. 235–265, 2005.
- [21] P. Ladevèze, F. Pled, and L. Chamoin, “[New bounding techniques for goal-oriented error estimation applied to linear problems](#),” *International Journal for Numerical Methods in Engineering*, 2012.
- [22] M. Rüter and E. Stein, “[Goal-oriented a posteriori error estimates in linear elastic fracture mechanics](#),” *Computer methods in applied mechanics and engineering*, vol. 195, no. 4, pp. 251–278, 2006.
- [23] O. A. González-Estrada, J. J. Ródenas, S. P. A. Bordas, M. Duflot, P. Kerfriden, and E. Giner, “[On the role of enrichment and statistical admissibility of recovered fields in a-posteriori error estimation for enriched finite element methods](#),” *Engineering Computations*, vol. 29, no. 8, pp. 814–841, 2012.

- [24] O. A. González-Estrada, E. Nadal, J. J. Ródenas, P. Kerfriden, S. P. A. Bordas, and F. J. Fuenmayor, “[Mesh adaptivity driven by goal-oriented locally equilibrated superconvergent patch recovery](#),” *Computational Mechanics*, vol. 53, no. 5, pp. 957–976, 2014.
- [25] F. Larsson, K. Runesson, and P. Hansbo, “[Computation of goal-oriented a posteriori error measures in space-time finite elements for viscoplasticity](#),” in *Trends in computational structural mechanics*, pp. 499–510, Int Center Numerical Mmethods Engineering, 2001.
- [26] S. Bordas and M. Duflot, “[Derivative recovery and a posteriori error estimate for extended finite elements](#),” *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 35, pp. 3381–3399, 2007.
- [27] M. Duflot and S. Bordas, “[A posteriori error estimation for extended finite elements by an extended global recovery](#),” *International Journal for Numerical Methods in Engineering*, vol. 76, no. 8, pp. 1123–1138, 2008.
- [28] W. Bangerth and R. Rannacher, “Finite element approximation of the acoustic wave equation: Error control and mesh adaptation,” *East West Journal of Numerical Mathematics*, vol. 7, no. 4, pp. 263–282, 1999.
- [29] W. Bangerth and R. Rannacher, “[Adaptive finite element techniques for the acoustic wave equation](#),” *Journal of Computational Acoustics*, vol. 9, no. 02, pp. 575–591, 2001.
- [30] R. Becker and R. Rannacher, “[An optimal control approach to a posteriori error estimation in finite element methods](#),” *Acta numerica*, vol. 10, no. 1, pp. 1–102, 2001.
- [31] W. Bangerth, M. Geiger, and R. Rannacher, “[Adaptive Galerkin finite element methods for the wave equation](#),” *Computational Methods in Applied Mathematics*, vol. 10, pp. 3–48, 2010.
- [32] G. R. Liu, K. Zaw, and Y. Y. Wang, “[Rapid inverse parameter estimation using reduced-basis approximation with asymptotic error estimation](#),” *Computer Methods in Applied Mechanics and Engineering*, vol. 197, no. 45, pp. 3898–3910, 2008.
- [33] C. Prudhomme, D. V. Rovas, K. Veroy, L. Machiels, Y. Maday, A. T. Patera, and G. Turinici, “[Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods](#),” *Journal of Fluids Engineering*, vol. 124, no. 1, pp. 70–80, 2002.
- [34] P. Chen and A. Quarteroni, “[Accurate and efficient evaluation of failure probability for partial different equations with random input data](#),” *Computer Methods in Applied Mechanics and Engineering*, vol. 267, pp. 233–260, 2013.
- [35] K. Urban, S. Volkwein, and O. Zeeb, “[Greedy sampling using nonlinear optimization](#),” in *Reduced Order Methods for modeling and computational reduction*, pp. 137–157, Springer, 2014.
- [36] M. Meyer and H. G. Matthies, “[Efficient model reduction in non-linear dynamics using the Karhunen-Loeve expansion and dual-weighted-residual methods](#),” *Computational Mechanics*, vol. 31, no. 1, pp. 179–191, 2003.
- [37] T. Bui-Thanh, K. Willcox, O. Ghattas, and B. van Bloemen Waanders, “[Goal-oriented, model-constrained optimization for reduction of large-scale systems](#),” *Journal of Computational Physics*, vol. 224, no. 2, pp. 880–896, 2007.
- [38] K. Willcox, O. Ghattas, B. van Bloemen Waanders, and B. Bader, “[An Optimization Frame work for Goal-Oriented, Model-Based Reduction of Large-Scale Systems](#),” in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC’05. 44th IEEE Conference on*, pp. 2265–2271, IEEE, 2005.

- [39] J. Nocedal and S. J. Wright, *Numerical optimization*, vol. 2. Springer New York, 1999.
- [40] M. Heinkenschloss, “[Numerical solution of implicitly constrained optimization problems](#),” *Rice University Department of Computational and Applied Mathematics Tech. Rep*, 2008.
- [41] B. Haasdonk and M. Ohlberger, “[Reduced basis method for finite volume approximations of parametrized linear evolution equations](#),” *ESAIM: Mathematical Modelling and Numerical Analysis*, vol. 42, no. 02, pp. 277–302, 2008.
- [42] J. L. Lions, *Optimal control of systems governed by partial differential equations*. Springer, 1971.
- [43] K.-J. Bathe, *Finite element procedures*. New Jersey, 1996.
- [44] T. J. R. Hughes, *The finite element method: Linear static and dynamic finite element analysis*. Englewood Cliffs, New Jersey: Prentice-Hall International, Inc, 1987.
- [45] W. J. T. Daniel, “[The subcycled Newmark algorithm](#),” *Computational Mechanics*, vol. 20, no. 3, pp. 272–281, 1997.
- [46] K. Zaw, G. R. Liu, B. Deng, and K. B. C. Tan, “[Rapid identification of elastic modulus of the interface tissue on dental implants surfaces using reduced-basis method and a neural network](#),” *Journal of biomechanics*, vol. 42, no. 5, pp. 634–641, 2009.
- [47] S. Volkwein, “[Proper Orthogonal Decomposition: Theory and Reduced-Order Modelling](#),” *Lecture Notes, University of Konstanz*, 2013.
- [48] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2005.
- [49] N. C. Nguyen, “[A multiscale reduced-basis method for parametrized elliptic partial differential equations with multiple scales](#),” *Journal of Computational Physics*, vol. 227, no. 23, pp. 9807–9822, 2008.
- [50] J. L. Eftang, D. J. Knezevic, and A. T. Patera, “[An hp certified reduced basis method for parametrized parabolic partial differential equations](#),” *Mathematical and Computer Modelling of Dynamical Systems*, vol. 17, no. 4, pp. 395–422, 2011.
- [51] B. Haasdonk, “[Convergence rates of the pod–greedy method](#),” *ESAIM: Mathematical Modelling and Numerical Analysis*, vol. 47, no. 03, pp. 859–873, 2013.
- [52] M. Yano, A. T. Patera, and K. Urban, “[A space-time certified reduced basis method for Burgers equation](#),” *Math. Mod. Meth. Appl. S.*, submitted, 2012.
- [53] M. Yano, “[A Space-Time Petrov–Galerkin Certified Reduced Basis Method: Application to the Boussinesq Equations](#),” *SIAM Journal on Scientific Computing*, vol. 36, no. 1, pp. A232–A266, 2014.
- [54] Y. Saad, *Iterative methods for sparse linear systems*. Siam, 2003.
- [55] S. Wang, G. R. Liu, K. C. Hoang, and Y. Guo, “[Identifiable range of osseointegration of dental implants through resonance frequency analysis](#),” *Medical engineering & physics*, vol. 32, no. 10, pp. 1094–1106, 2010.
- [56] T. Rabczuk, S. Bordas, and G. Zi, “[A three-dimensional meshfree method for continuous multiple-crack initiation, propagation and junction in statics and dynamics](#),” *Computational mechanics*, vol. 40, no. 3, pp. 473–495, 2007.
- [57] J.-H. Song, H. Wang, and T. Belytschko, “[A comparative study on finite element methods for dynamic fracture](#),” *Computational Mechanics*, vol. 42, no. 2, pp. 239–250, 2008.

- [58] T. Rabczuk, S. Bordas, and G. Zi, “On three-dimensional modelling of crack growth using partition of unity methods,” *Computers & Structures*, vol. 88, no. 23, pp. 1391–1411, 2010.
- [59] H. Nguyen-Vinh, I. Bakar, M. Msekh, J.-H. Song, J. Muthu, G. Zi, P. Le, S. Bordas, R. Simpson, S. Natarajan, *et al.*, “Extended finite element method for dynamic fracture of piezo-electric materials,” *Engineering Fracture Mechanics*, vol. 92, pp. 19–31, 2012.
- [60] N. Nguyen-Thanh, T. Rabczuk, H. Nguyen-Xuan, and S. Bordas, “An alternative alpha finite element method (A α FEM) for free and forced structural vibration using triangular meshes,” *Journal of computational and applied mathematics*, vol. 233, no. 9, pp. 2112–2135, 2010.

A Non-dimensionalization of damped wave equation (weak form)

In the following, the physical and non-dimensional terms will be denoted as \bar{x} and x , respectively. With all the terms defined as in (3)–(4) and assuming zero mass-proportional Rayleigh damping coefficients, the weak form of the dimensional damped wave equation considered in this work has the following form: $\forall \bar{t} \in [0, \bar{T}]$ and a test function \bar{v}

$$\int_{\Omega} \bar{\rho} \bar{v}_i \frac{\partial^2 \bar{u}_i}{\partial \bar{t}^2} + \bar{\beta} \frac{\partial}{\partial \bar{t}} \int_{\Omega} \frac{\partial \bar{v}_i}{\partial \bar{x}_j} \bar{C}_{ijkl} \frac{\partial \bar{u}_k}{\partial \bar{x}_l} + \int_{\Omega} \frac{\partial \bar{v}_i}{\partial \bar{x}_j} \bar{C}_{ijkl} \frac{\partial \bar{u}_k}{\partial \bar{x}_l} = \int_{\Gamma_1} \bar{v}_i \bar{\phi}_i \quad \text{on } \bar{\Omega}, \quad (30)$$

which subjects to the boundary conditions

$$\bar{u}_i = \bar{u}_i^d \quad \text{on } \bar{\Gamma}_D, \quad (31a)$$

$$\bar{\sigma}_{ij} n_j = \bar{C}_{ijkl} \left(\frac{\partial \bar{u}_k}{\partial \bar{x}_l} + \bar{\beta} \frac{\partial}{\partial \bar{t}} \frac{\partial \bar{u}_k}{\partial \bar{x}_l} \right) n_j = \bar{\phi}_i \quad \text{on } \bar{\Gamma}_1, \quad (31b)$$

and initial conditions

$$\bar{u}_i(\bar{t} = 0) = \bar{u}_i^{d,0} \quad \text{on } \bar{\Omega}, \quad (32a)$$

$$\frac{\partial \bar{u}_i(\bar{t} = 0)}{\partial \bar{t}} = \bar{v}_i^{d,0} \quad \text{on } \bar{\Omega}. \quad (32b)$$

Let L^* , ρ^* and E^* are correspondingly the characteristic length, mass density and stress of the model and define the non-dimensional variables as

$$x_i = \frac{\bar{x}_i}{L^*}, \quad u_i = \frac{\bar{u}_i}{L^*}, \quad \phi_i = \frac{\bar{\phi}_i}{E^*}, \quad (33a)$$

$$C_{ijkl} = \frac{1}{E^*} \bar{C}_{ijkl}, \quad t = \sqrt{\frac{E^*}{\rho^* L^{*2}}} \bar{t}, \quad \beta = \sqrt{\frac{E^*}{\rho^* L^{*2}}} \bar{\beta}. \quad (33b)$$

Under this transformation, the non-dimensional weak form of (30) is defined as: $\forall t \in [0, T]$, $T = \sqrt{\frac{E^*}{\rho^* L^{*2}}} \bar{T}$ and a test function v

$$\int_{\Omega} v_i \frac{\partial^2 u_i}{\partial t^2} + \beta \frac{\partial}{\partial t} \int_{\Omega} \frac{\partial v_i}{\partial x_j} C_{ijkl} \frac{\partial u_k}{\partial x_l} + \int_{\Omega} \frac{\partial v_i}{\partial x_j} C_{ijkl} \frac{\partial u_k}{\partial x_l} = \int_{\Gamma_1} v_i \phi_i \quad \text{on } \Omega, \quad (34)$$

with boundary conditions

$$u_i = \frac{\bar{u}_i^d}{L^*} \quad \text{on } \Gamma_D, \quad (35a)$$

$$\sigma_{ij}n_j = C_{ijkl} \left(\frac{\partial u_k}{\partial x_l} + \beta \frac{\partial}{\partial t} \frac{\partial u_k}{\partial x_l} \right) n_j = \frac{\bar{\phi}_i}{E^*} \quad \text{on } \Gamma_1, \quad (35b)$$

and initial conditions

$$u_i(t=0) = \frac{\bar{u}_i^{\text{d},0}}{L^*} \quad \text{on } \Omega, \quad (36a)$$

$$\frac{\partial u_i(t=0)}{\partial t} = \sqrt{\frac{\rho^*}{E^*}} \bar{v}_i^{\text{d},0} \quad \text{on } \Omega. \quad (36b)$$